



# **Transforming System Engineering through Model-Centric Engineering**

**Technical Report SERC-2015-TR-109**

**November 18, 2015**

**PI:** Dr. Mark Blackburn, Stevens Institute of Technology

## **Research Team**

Dr. Mary Bone, Stevens Institute of Technology

Dr. Gary Witus, Wayne State University

## **Sponsor**

NAVAIR, DASD (SE)

Copyright © 2015 Stevens Institute of Technology, Systems Engineering Research Center

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Systems Engineering Research Center (SERC) under Contract H98230-08-D-0171 (Task Order 041, RT 141). SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

#### NO WARRANTY

THIS STEVENS INSTITUTE OF TECHNOLOGY AND SYSTEMS ENGINEERING RESEARCH CENTER MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. STEVENS INSTITUTE OF TECHNOLOGY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. STEVENS INSTITUTE OF TECHNOLOGY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

This material has been approved for public release and unlimited distribution.

## TABLE OF CONTENTS

---

<b>Acknowledgments .....</b>	<b>vii</b>
<b>Executive Summary.....</b>	<b>1</b>
<b>Introduction .....</b>	<b>5</b>
<b>Objective .....</b>	<b>5</b>
<b>Scope .....</b>	<b>6</b>
<b>Organization of Document .....</b>	<b>7</b>
<b>Research Summary .....</b>	<b>8</b>
<b>Modeling Terminology and Model Lexicon Status .....</b>	<b>8</b>
<b>Changing Nature of Engineering and the Impact on Systems Engineering.....</b>	<b>9</b>
<b>Current State of Model Centric Engineering .....</b>	<b>10</b>
Operational Perspective of Model-Centric Integration .....	10
Virtual Verification and Validation Continuum .....	11
Mission-Level Simulation Integration with System Simulation and digital Assets.....	12
3D Environments and Visualization .....	13
Dynamic Operational Views for Mission and System Simulation and Analysis .....	13
Multidisciplinary Design, Analysis and Optimization (aka Tradespace Analysis).....	14
Interactive Successful Refinement of Design Space.....	15
1D, 2D, & 3D Model Creation, Simulation, Analysis for Physics-based Design.....	16
Modeling and Simulation Integration with Embedded System Code .....	17
Platform-based Approaches.....	17
Modeling and Simulation Reducing Physical Crash Testing .....	18
Modeling and Simulation of the Manufacturing.....	18
Workflow Automation to Subsume Process .....	18
Single Source of Truth, Modeling Patterns, Ontologies, Profiles, Continuous Integration .....	19
Quantification of Margins and Uncertainty .....	20
<b>Gaps and Challenges .....</b>	<b>20</b>
Cross-Domain Interoperability.....	20
Complexity of Software and the Verification Challenge .....	21
Metrics and Tools for Verification and Validation of Cyber-Physical System .....	23
<b>Summary .....</b>	<b>24</b>
<b>Part II: Previous Work Material .....</b>	<b>27</b>
<b>Task 1 - Assessing the State-of-the-Art MBSE .....</b>	<b>27</b>
<b>Discussion Narratives and Measurement Summary.....</b>	<b>28</b>
<b>Task 1 - Process.....</b>	<b>30</b>
<b>Scenario Collection .....</b>	<b>30</b>
Organizational Type .....	31
Organizational Scope.....	31
Factors Definition Example .....	32
<b>Discussion Summaries .....</b>	<b>33</b>
<b>Predictive Model.....</b>	<b>33</b>
Rationale for Bayesian Networks .....	34
Data - Likert Scales (Ranked Scales) .....	35
<b>Task 2 - Common Model Lexicon .....</b>	<b>35</b>

Ontology vs. Lexicon .....	35
Tool for Representing Word Relationships .....	36
The Lexicon.....	36
Sources of Information.....	37
Web Presentation .....	38
<b>Task 3 - Modeling the Vision and Relating to the “As Is” and Airworthiness Process.....</b>	<b>42</b>
Status of “As Is” and Airworthiness Process Modeling.....	43
What is a model? .....	44
Vision Model Context.....	46
Containing System.....	48
Designing System .....	49
Operational Perspectives on a Vision Concept .....	50
Scope to Program of Record through Digital Critical Design Review .....	53
Context for Program of Record System .....	54
Structuring Mission-Level Analysis.....	55
Modeling and Tools for the Vision .....	58
Model-Centric Engineering Environment Perspectives .....	58
Model Transformation Rather Than Model Evolution .....	59
Crossing the Virtual “V” by Leveraging Models, Digital and Physical Surrogates .....	60
Decision Framework.....	60
Systems Engineering Technical Review (SETR) Manager .....	61
Integrated MCE Environment.....	61
<b>Task 4 – Integrated Framework for Risk Identification and Management .....</b>	<b>63</b>
Risk Context.....	63
Risk of Consequence from Model Centric Engineering Transformation.....	64
Future Root Causes .....	65
Scope of the Risk Framework .....	66
Modeling and Methods for Uncertainty Quantification .....	68
Dakota Sensitivity Analysis and Uncertainty Quantification (UQ) .....	68
An Overview of Quantification of Margins and Uncertainty.....	70
Risk Framework Approach to Uncertainty Modeling and Prediction .....	72
Predictive Models for Risk.....	73
Risk Framework Captures Knowledge.....	74
Model Validation and Simulation Qualification .....	75
Improving the Integrity of Models .....	76
Model-Centric Methods and Tools Enable Approaches to Safety and Airworthiness.....	77
MCE Methods to Generate Failure Modes and Effects Analysis (FMEA) models .....	78
Mission-level Model-based Safety Analysis .....	79
Risk Reliance on Model Correctness and Accuracy.....	80
Error and Uncertainty Source and Propagation Analysis .....	81
Risk in a Collaborative Environment .....	82
Risk Related Research .....	83
<b>Part II Summary .....</b>	<b>84</b>
<b>Appendix A: Factor Definitions .....</b>	<b>85</b>
<b>Acronyms and Abbreviation .....</b>	<b>88</b>

<b>Trademarks .....</b>	<b>91</b>
<b>References.....</b>	<b>93</b>

## FIGURES

---

Figure 1. Four Tasks to Assess Technical Feasibility of “Doing Everything with Models” .....	1
Figure 2. Model Centric Provides Digital Integration Between Views .....	11
Figure 3. Dynamic Models and Surrogates to Support Continuous “Virtual V&V” Early in the Lifecycle ..	12
Figure 4. Cave Automated Virtual Environment (CAVE) .....	13
Figure 5. Dynamic OV1 with Integrations to Other Models and Digital Assets .....	14
Figure 6. MDAO Compared with Legacy Tradespace Analysis [51] .....	15
Figure 7. DARPA META Concept .....	16
Figure 8. Vehicle System Platform-based Model.....	17
Figure 9. DARPA META Program .....	22
Figure 10. META Program Claims Conventional V&V Techniques do not Scale to Highly Complex Systems .....	23
Figure 11. Traceability and Scope of Data Collection for Task 1.....	25
Figure 12. Engineering Resilient System Conceptual Representation of Environment [63] - Enhanced....	26
Figure 13. Measurement Collection Instrument.....	28
Figure 14. Collection Instrument Results .....	30
Figure 15. Spreadsheet Instrument Collection .....	31
Figure 16. Bayesian Network Underlying Collection Instrument.....	34
Figure 17. Sample Graphic Representation from Ontological Software.....	36
Figure 18. Published Web Page from Data Collection Spreadsheet .....	39
Figure 19. Model Representation and Lexicon Tree .....	40
Figure 20. Partial Graphical Representation of Lexicon.....	41
Figure 21. Tabular Representation of Lexicon .....	42
Figure 22. Two Model Views: Mobile and Relative Size of Planets .....	45
Figure 23. Putting the Vision into Context.....	48
Figure 24. Perspectives on Aspects of the Containing System .....	49
Figure 25. Overarching Concept for Vision .....	53
Figure 26. Mission Context for System Capability .....	55
Figure 27. NASA/JPL Architecture Framework Tool (AFT) for Architecture Description .....	57
Figure 28. Integrated Environment for Iterative Tradespace Analysis of Problem and Design Space .....	62

Figure 29. Dakota Framework Integration Wraps User Application.....	69
Figure 30. Example for Understanding Margins and Uncertainty .....	69
Figure 31. Pulling Together Concept Associated with QMU .....	72
Figure 32. Bayesian Model Derived from Airworthiness Factors .....	74
Figure 33. Example Failure Modes and Effects Analysis Metamodel .....	81

## TABLES

---

Table 1. Initial Lexicon Capture Tool.....	37
Table 2. Discussion Instrument Factor Definition.....	85

## ACKNOWLEDGMENTS

---

We wish to acknowledge the great support of the NAVAIR sponsors and stakeholders, including stakeholders from other industry partners that have been very helpful and open about the challenges and opportunities of this promising approach to transform systems engineering.

We want to specifically thank Dave Cohen who established the vision for this project, and our NAVAIR team, Jaime Guerrero, Gary Strauss, Brandi Gertsner and Ron Carlson, who has worked closely on a weekly basis in helping to collaboratively research this effort. We want to thank Dennis Reed and Howard Owens who participated in some of the organizational discussions.

We also want to thank all, currently more than 180 stakeholders that participated in over 30 organizational discussion and 15 working session, including some from industry that will remain anonymous in recognition of our need to comply with proprietary and confidentiality agreements associated with Task 1.

As there are so many stakeholders that supported this effort, we wish to recognize them all. We sincerely apologize if we have missed anyone else that has supported our efforts.

Andrew Devine	Doreen Dixon	John McKeown	Philomena Zimmerman
Art Pyster	Eirik Hole	Joseph Tolarski	Richard Price
Aumber Bhatti	Eric (Tre') Johnsen	Judith Dahmann	Richard Yates
Bill Brickner	Fatma Dandashi	Keith Carter	Rob Cloutier
Bhawana Sharma	Fran Chamberlain	Leslie Taylor	Sara Shaddix
Brad Kidwell	Geethesh Kukkala	Larry Smith	Scott Lucero
Brian Nolan	Jae Pfeffer	Lisette Fortuno	Shahram Bavani
Brent Gordon	James Carroll	Michael Gaydar	Stan Rifkin
Bob Levitt	James Light	Michael Alcantara	Stu Young
Chris Owen	Jeff Smallwood	Mike Kelley	Todd Standard
Dale Moore	John Funk	Paul Montgomery	Tom Blakely

## EXECUTIVE SUMMARY

This is the final report of the Systems Engineering Research Center (SERC) research task RT-141 that finalizes the related tasks under RT-48/118. These RTs focused on a Vision held by NAVAIR's leadership to assess the **technical feasibility** of a **radical transformation** through a more holistic model-centric engineering approach. The expected capability of such an approach would enable mission-based analysis and engineering that reduces the typical time by at least 25 percent from what is achieved today for large-scale air vehicle systems. The effort investigates the technical feasibility of moving to a "complete" model-centric lifecycle and includes four overarching and related tasks as shown in Figure 1. These tasks include:

- Task 1: Surveying Industry, Government and Academia to understand the state-of-the-art of a holistic approach to model-centric engineering ("everything digital")
- Task 2: Develop a common lexicon for things related to models, including model types, levels, uses, representation, visualizations, etc.
- Task 3: Model the "Vision," but also relate it to the "As Is" and Airworthiness processes
- Task 4: Integrate a Risk Management framework with the Vision

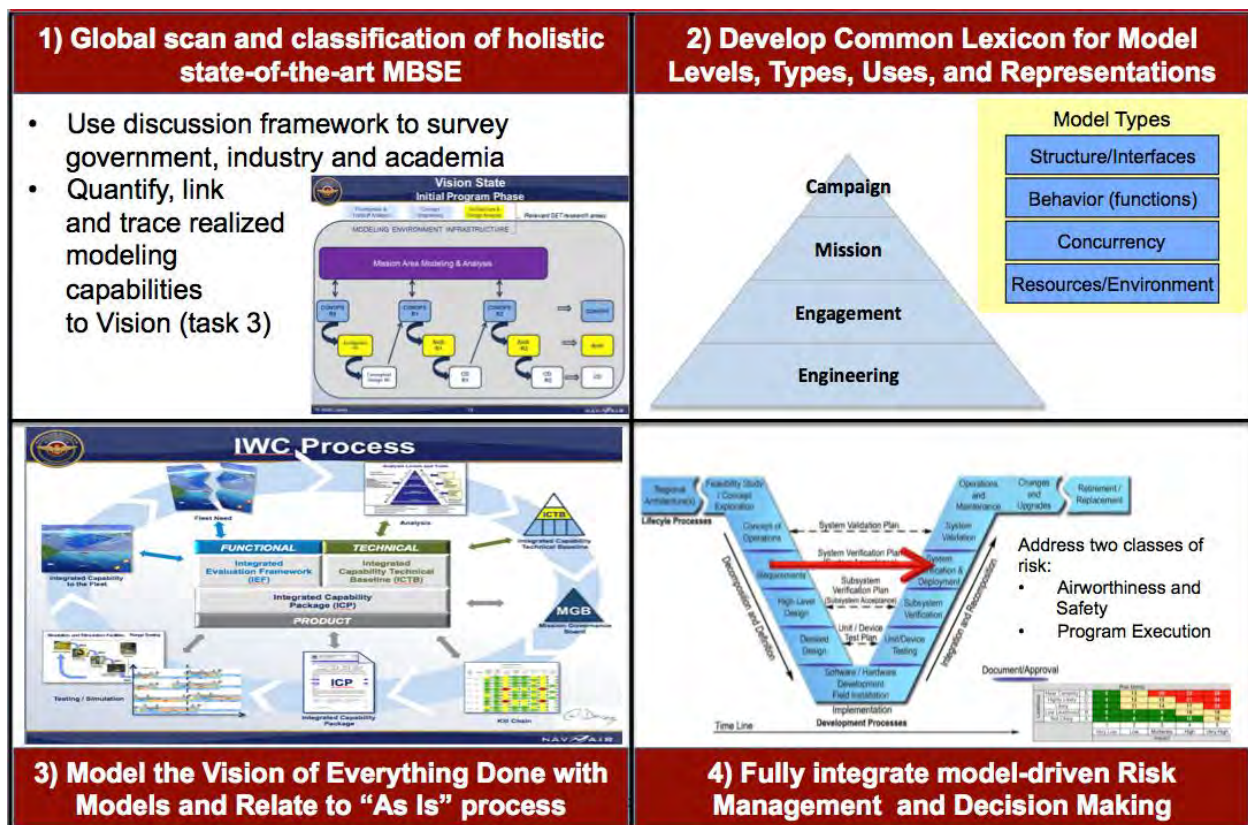


Figure 1. Four Tasks to Assess Technical Feasibility of "Doing Everything with Models"

There has been considerable emphasis on understanding the state-of-the-art through discussions with industry, government and academia. We have conducted over 29 discussions, including 21 on site, and 15 working sessions, as well as several follow-up discussions on some of the identified challenge areas. We did not do a survey, but rather had open-ended discussions. We asked the meeting coordinators to in general:



Tell us about the most advanced and holistic approach to model-centric engineering you use or have seen used.

The spectrum of information was very broad; there really is no good way to make a comparison. In addition, we had proprietary information agreements with most industry organizations. The objective was not to single out any specific organization, therefore, we will summarize, in the aggregate, what we heard in this report as it relates to the NAVAIR research objective.

Our research suggests that model-centric engineering is in use and adoption seems to be accelerating. Model-centric engineering can be characterized as an overarching digital engineering approach that integrates different model types with simulations, surrogates, systems and components at different levels of abstraction and fidelity across disciplines throughout the lifecycle. Industry is trending towards more integration of computational capabilities, models, software, hardware, platforms, and humans-in-the-loop. The integrated perspectives provide cross-domain views for rapid system level analysis allowing engineers from various disciplines using dynamic models and surrogates to support continuous and often virtual verification and validation for tradespace decisions in the face of changing mission needs.

Enabling digital technologies are changing how organizations are conceptualizing, architecting, designing, developing, producing, and sustaining systems and systems of systems (SoS). Some use model and simulation environments for customer engagements, as well as design engineering analyses and review sessions. While they do use commercial technologies, most have been innovating and have developed a significant amount of enabling technology – some call it their “secret sauce.” The research findings and recommendations are based on seeing demonstrations and evidence of cross-cutting technologies and methods. Demonstrations have included mission-level simulations that are being integrated with system simulation, digital assets and aircraft products providing cloud-like services enabled by the industrial Internet. There have been demonstrations of 1D, 2D, and 3D modeling and simulations with a wide array of solvers and visualization capabilities. We have also been in an immersive Cave Automated Virtual Environment. We have seen the results of platform-based approaches directly focused on speed-to-market, and more.

The analysis of captured evidence in this research suggests that there is a transition from model-based engineering to model-centric engineering. The advances and availability of high performance computing, capabilities to provide cross-domain and multi-physics model integration, and methods and tools to assess model integrity will support the need for reducing the time to deliver system capabilities. Even sociotechnical computing is enabling new ways to access and more transparently collaborate and share information, and it can be a key contributor to a radical transformation to model-centric engineering.

## **Findings**

The findings conveyed to NAVAIR leadership definitely indicate that it is technically feasible to transform systems engineering at NAVAIR similar to the transformation seen across large organizations in aerospace, automotive, and government. This transformation increases the likelihood of achieving at least 25 percent reduction in acquisition. A summary of the data analysis is presented in a traceability matrix that captured 21 topic-discussion areas summarized in this report. The matrix also provided evidence of traceability to different instances of organizational use and their possible impacts/relationships on characteristics, such as: performance, integrity, affordability, risk, methodologies, and within a single source of technical truth<sup>1</sup>.

---

<sup>1</sup> Our sponsor uses the term single source of technical truth; others have used the phrases such as: single source of truth, single authoritative representation of the system. Any of these terms apply to this concept.

A rule of thumb is that the effort/time to get from Milestone A to Critical Design Review (CDR) is about 30 percent of the total time, where the time from CDR to Initial Operating Capability (IOC) is about 70 percent of the total time. With some of the new approaches to produce digital information, which considers modeling and simulation analysis of manufacturability prior to CDR, the digital information at CDR could significantly reduce the 70 percent effort from CDR to IOC, which also builds to the argument for being able to reduce the acquisition time by 25 percent with MCE.

The feasibility of Systems Engineering transformation through MCE has three key critical technical items: 1) cross-domain and multi-physics model integration, 2) ensuring model integrity (trust in the model predictions), and 3) high performance computing, which is an enabler for 1 and 2, but critical due to the scale and complexity of next generation systems.

There are a number of examples that span various domains across aerospace, automotive, and involving commercial, government and academic organizations. Many have lessons learned and examples covering a number of themes spanning technologies, methods, and usage at various stages of the lifecycle, even taking into consideration constraints for manufacturability in design-space exploration. Therefore MCE is not necessarily the catalyst; rather it is enabled by approaches that support data-driven decision-making that will subsume processes through:

- Single Source of Technical Truth (SSTT) – one source of information
- Views and viewpoints for the multidisciplinary stakeholders into the SSTT
- Multidisciplinary Design, Analysis and Optimization (MDAO) in both tradespace exploration and analysis of the problem and design space
- Workflow orchestration – by having the data dependencies being semantically linked within the SSTT
- Enabled by High Performance Computing (HPC)

## **Recommendation**

NAVAIR senior leadership confirmed that the research finding and analysis have validated their vision hypothesis stated at the System Engineering Transformation kickoff meeting of RT-48. They conclude that NAVAIR must move quickly to keep pace with the other organizations that have adopted MCE and who continue to evolve at an accelerating pace enabled by the advances in technologies and improved methods. NAVAIR must also transform in order to continue to perform effective oversight of weapon system development by primes that are using modern modeling methods for system development. The risks of not moving forward include making acquisition decisions with progressively less technical-truth insight and the proliferation of disparate, redundant and stove-piped data and models, and lacking MCE capabilities and knowledge needed to understand an increasingly complex problem and design space.

The path forward has challenges but also many opportunities, both technical and sociotechnical. It must include a modeling framework with HPC that enables SSTT, integration of multi-domain and multi-physics models, and provides for a method for model integrity. The modeling and infrastructure for a digital engineering environment is a critical step to enable a SSTT. While there are literally thousands of tools, they are often federated and there is no one single solution that can be purchased. Every organization providing inputs to this research has had to architect and engineer their model-centric engineering environment, most have selected commercial tools and have developed the integrating fabric between the different tools, models, and data. This approach often uniquely positions them with some advantages among the rest. Some organizations have encoded historical knowledge in reference models, model patterns to embed methodological guidance to support continuous orchestration of analysis through new modeling metrics, automated workflow, and more. The items to investigate further include but are not limited to:

- Cross-domain integration of models to address the heterogeneity of the various tools and environments
- Model integrity to ensure trust in the model predictions by understanding and quantifying margins and uncertainty
- Modeling methodologies that can embed demonstrated best practices and provide computational technologies for real-time training within digital engineering environments
- Multidisciplinary System Engineering transformation roadmap that looks across:
  - Technologies and their evolution
  - How people interact through digitally enabled technologies and new needed competencies
  - How methodologies enabled by technologies change and subsume processes
  - How acquisition organizations and industry operate in a digital engineering environment throughout the phases of the lifecycle (including operations and sustainment)
  - Governance within this new digital and continually adapting environment

This report aggregates information contained in the final technical reports of RT-48 and RT-118 so that readers can get the key information from this report. The report is structured so that the key findings and next steps are described in the first section. The report then provides updated clarification on the scope given by our NAVAIR sponsor. Part II section provide additional detail to summarize the efforts that are aligned with tasks 1 through 4.

## INTRODUCTION

---

In 2013, the Naval Air Systems Command (NAVAIR) at the Naval Air Station, Patuxent River, Maryland initiated a research task (RT-48) to assess the technical feasibility of creating/leveraging a more holistic Model-Based Systems Engineering (MBSE) approach to support mission-based analysis and engineering in order to achieve a 25 percent reduction in development time from that of the traditional large-scale air vehicle weapon systems. The research need was focused on the evaluation of emerging system design through computer (digital) models. The first phase of the effort under RT-48 created a strategy and began collecting and structuring evidence to assess the technical feasibility of moving to a “complete” model-driven lifecycle. The second phase conducted under RT-118 involved an extensive outreach to understand the state-of-the-art in using models. The third phase under RT-141 conducted additional discussions on leading-edge approaches and correlated the analysis to provide evidence that traced to organizational examples needed to finalize the finding and recommendations.

A goal is to leverage virtual designs that integrate with existing systems data and simulations, as well as surrogates at varying levels of refinement and fidelity to support a more continuous approach to mission and systems analysis and design refinement. What emerged as Model-Centric Engineering (MCE) can equally well be characterized as Digital Engineering. This broader view of the use of models has moved our team to use the term model-centric engineering. MCE can be characterized as an overarching digital approach for integrating different model types with simulations, surrogates, systems and components at different levels of abstraction and fidelity, including software-, hardware-, platform-, and human-in-the-loop across disciplines throughout the lifecycle. The expanse of the organizational discussions even included modeling and simulation to assess the manufacturability of a design during tradespace analysis of the designs, and therefore we use the term MCE in the broadest way in this report.

The larger context of the NAVAIR mission seeks a radical transformation in the way they operate within NAVAIR and with industry through MCE. The Vision of NAVAIR is to establish an environment to evaluate the emerging system design through computer models and demonstrate system compliance to user performance and design integrity requirements, while managing airworthiness risks. It is anticipated that this model-centric approach can streamline and radically transform the traditional document-centric process that decomposes requirements and their subsequent integrated analysis, which is currently aligned with the Department of Defense (DoD) systems engineering V-model (i.e., the “V”). By providing more tightly coupled and dynamic linkages between the two sides of the traditional “V,” more efficient and focused requirements decomposition would eliminate thousands of pages of documentation delivered via contract data requirements that now substitute for directly invoking, manipulating, and examining the design through computer-based models and digital artifacts.

---

## OBJECTIVE

The objectives characterized by the sponsor at the RT-48 kickoff meeting were re-stated at the out briefing to senior leadership at NAVAIR to ensure that the research covered the key objectives; those objectives included:

- Include both models to assess “performance” and models for assessing “integrity” such as:
  - Performance: aero, propulsion, sensors, etc.
  - Integrity: Finite Element Analysis (FEA), Computation Fluid Dynamics (CFD), reliability, etc. – can we build it, can we trust it
  - A stated challenge was: how can “integrity” be accomplished when the current situation involves federations of models that are not integrated?

- Continuous hierarchical and vertical flow enabled by models and iterative refinement through tradespace analysis, concept engineering, and architecture and design analysis
- Integrated mission area model analysis supported by a modeling infrastructure starting at pre-milestone A
- Continuous and iterative refinement of digital (model) versions representing digital characterizations of the information for Preliminary Design Review (PDR)
  - Analyzed with integrated performance and integrity capabilities
  - Integrated with bidirectional vertical flows between mission-level modeling and assessment, and the system(s) of interest (Program of Record)
  - Refinement of lower-fidelity PDR models to higher-fidelity Critical Design Review (CDR) models characterizing the incrementally refined Concept of Operations (CONOPS), architecture, and detailed design and constraints
  - Models-to-manufacturability and models-to-training were desired

Therefore, there are three overarching research questions:

1. Is it **technically feasible** to use model-centric engineering in order to achieve at least a **25 percent reduction in the time** it takes to deliver a large-scale air vehicle weapon system?

There is a corollary to first question:

2. If we are going to rely more heavily on model-centric engineering, with an increasing use of modeling and simulations, how do we know that models/simulations used to assess “performance” have the needed “integrity” to ensure that the performance predictions are accurate?

While model-centricity may enable improved automation and greater efficiency, NAVAIR seeks its use in an improved operational paradigm, and therefore the third questions is:

3. Can we **radically transform** the way that NAVAIR and all contributing stakeholders operate in conceptualizing, architecting, designing, developing, producing, and sustaining systems and SoS?

Our sponsor in the original kickoff briefing stated:

- “Blow up” the current “Newtonian” approach and move to a “Quantum” approach that recognizes and capitalizes on current and emerging trends and enabling technologies

It is acknowledged that there are many possible hurdles beyond technical feasibility (e.g., organizational adoption, training, usability, etc.), but they have in general been reduced in priority for this phase of the effort. The path forward includes a road map to factor in some of these other considerations.

The focus of the research task is scoped at the system level, often characterized as the Program of Record (POR) plus weapons, for an air vehicle system, although the next steps are directed to include Systems of Systems (SoS). The timeframe for the technical feasibility is scoped at 10 years. This would reflect on the “Vision” concept, but also the operational aspects of government organizations like NAVAIR interacting in a more continuous type of collaboration with industry stakeholders.

---

## SCOPE

Given the objectives above, we were directed to scope the effort to focus on the front-end of the lifecycle from pre-milestone A to critical design review (CDR). This is typically considered the front half of the “V” model. However, as is discussed later in this report, many of our meeting discussions went well beyond

this scope. We do document most of these potential out-of-scope ideas as they may play a role in a radical transformation.

---

## ORGANIZATION OF DOCUMENT

**Section 1** provides an overview of the research objectives, scope and organization of this report.

**Section 2** provides the summary of our efforts, findings, analysis and recommendations including key aspects from RT-48/118.

In order to be comprehensive, we are including discussions of the four tasks in a manner consistent with the final technical report from RT-48/118 [20] [22].

**Section 3** describes the approach for having discussions with commercial, government, and academic organizations in order to assess the most holistic state-of-the-art use or vision for model-centric engineering.

**Section 4** describes the approach for developing a model lexicon to characterize such things as: model levels, model types, model uses, representations, and other categories related to “models.”

**Section 5** discusses the scope and concept of the Vision model and its relationship to the “As Is” artifacts and process that are currently in place for developing NAVAIR air vehicle systems; this also includes the airworthiness process. This section provides some context about the scope of the effort and the relationship to the NAVAIR mission-level modeling and simulation.

**Section 6** discusses a framework for risk identification and management that is primarily focused on how airworthiness and safety risk can be integrated in the Vision model. This section discusses classes of risks that we need to deal with in the context of model-centric engineering, tools and methods for quantification of margins and uncertainty, model validation, verification, accreditation, and simulation qualification.

**Section 7** provides some conclusions with a brief summary of the planned next steps.

There is additional backup information, including the factor definitions associated with the discussion collection instrument, a list of acronyms and abbreviations following the conclusion.

## RESEARCH SUMMARY

---

This section provides a summary of the findings, analysis, examples and recommendations of this research as the final deliverable for the four tasks discussed under RTs-48/118/141. This section presents the following information:

- Model-centric engineering terminology
- Model lexicon status
- Change nature of system engineering needed for Cyber Physical Systems
- 15 perspectives identified as themes from visits to, and discussions with industry, government, and academia to seek out the most advanced holistic uses of model-centric engineering
- Discussion about some challenges areas
- Summary and next steps

---

### MODELING TERMINOLOGY AND MODEL LEXICON STATUS

Modeling terminology can be confusing, and we created a model lexicon (Task 2). However, a simple definition is not always adequate as there are many overlapping definitions. Some of the terms are overloaded, and some with multiple definitions. The Lexicon may also need to be tailored to the needs of the organization and account for different definitions within different parts of the organization. While we did give some references and example uses in our lexicon, they do not necessarily completely convey the broad concepts such as model-centric engineering.

Status Task 2: we have captured over 750 named lexicon items related to the term “model,” including levels, types, uses, representations, standards, etc. There were many contributors to the lexicon. It has been reviewed several times, and updated based on the comments of the reviewers. We have delivered these model-lexicon artifacts to NAVAIR for them to post internally.

The directive when we started this research task was to investigate the most advanced approaches to Model-Based Systems Engineering (MBSE). However, during our organizational discussions some of the people we interacted with thought this to be Model-Based Engineering (MBE), others used the term Model-Centric Engineering [111], Integrated Model-Centric Engineering [8], Interactive Model-Centric Engineering [101], Model-Driven Development, Model-Driven Engineering (MDE) [13], and even Model-Based Enterprise [81], which brings in more focus on manufacturability. The concept characterized as Digital Thread<sup>2</sup> envisions a frameworks that merges physics-based models generated by the discipline engineers during the detailed design process with MBSE’s conceptual and top-level architectural models, resulting in a single authoritative representation of the system [121]. The words Digital Engineering have been used, because the nature of systems engineering is expanding to include more rigorous uses of models and cross-domain integration.

---

<sup>2</sup> Digital Thread: “An extensible, configurable and component enterprise-level analytical framework that seamlessly expedites the controlled interplay of authoritative technical data, software, information, and knowledge in the enterprise data-information-knowledge systems, based on the Digital System Model template, to inform decision makers throughout a system's life cycle by providing the capability to access, integrate and transform disparate data into actionable information.”

Systems have been categorized as: Physical, Computational, and Cyber Physical Systems (CPS) [116]. While the research did not attempt to focus on a particular type of system, invariably the discussions involved CPS. The phrase “cyber-physical systems” was coined by Helen Gill [55] and defined as physical, biological, and engineered systems whose operations are integrated, monitored, and/or controlled by a computational core. Components are networked at every scale. Computing is deeply embedded into every physical component, possibly even into materials. The computational core is an embedded system, usually demands real-time response, and is most often distributed.” CPSs are integrations of computation, networking, physical processes and often involving humans-in-the-loop. Many advanced systems have computational intelligence enabling self-adaptation and flexible autonomy.

CPS such as aircraft, automobiles, robotics, and space systems involve multi-discipline, multi-physics, and multi-vendors during conceptualization of the problem space, architecting the design space, generation and composability during implementation, and manufacturability of these systems. The magnitude of the tradespace is near infinite and the document-centric tools and methods for engineering systems today cannot address even complicated systems, much less complex systems. The impact of decisions regarding one design parameter on other design parameters is difficult, at best, to determine and can hamper effective trade space analysis. MCE provides some of the needed support by integrating different model types and tools for the simulation and analysis of systems and components at various levels of abstraction and fidelity across disciplines and throughout the lifecycle [120].

Some of the sponsor leadership sees less complex technology development outside of systems engineering like the mobile app phenomenon, and they have asked, “Why is this (MCE on the scale of large scale aircraft, e.g., the Joint Strike Fighter) so complicated?” Engineers and their customers live in a world full of consumer technology with powerful computing ability available in almost every aspect of their lives (vehicles, health care, smart homes, smart phones, etc.). This availability of connectedness in everyday life that streamlines personal needs creates desired environments within the workplace. Electronic records, calendars, forms, databases that are easy to search and link in commercial cloud environments are desired by engineers over paper documentation that is viewed as hard to locate and link to other artifacts. In daily life if a person decides they would like to be able to link, monitor, or find something they simply go the application store find an “app” and their need is met. The problem is that these are primarily in the computational domain, and not cross-domain or multi-physics like CPS.

Customers and engineers desire a more seamless flow of system development without duplicate or overlapping effort. System developers are looking for ways to generate digital threads extracted from the single source of technical truth. These digital threads would allow system developers to trace Concept of Operations (CONOPS) through tradespace analyses, to requirements all the way through the development process and reduce the number of regeneration and duplication of material through out the development process. A single source of truth could allow for easier recognition of impacts due to design and requirement changes, better indicators of program status (both technical and management), and earlier recognition of design/requirement anomalies or issues.

While MCE is a newer concept with limited measurable data, the state of MBSE has been tracked for some time. The perceived value of MBSE from practitioners in a 2015 study showed a 4.26 rating on a scale from one to five (with five being high value) [37]. This study showed also overtime this perceived value has increased from 4.05 in 2012 to the 4.26 in 2015. The study did not show why the perceived value has increased but it is theorized it could relate to better tools and more people able to execute MBSE well due to continued training and practicing. While MBSE is a part of the MCE it does not encompass the full idea and enabling technologies of MCE. Traditionally, MBSE has had limited use of



static modeling tools often focused mostly on architecture structure. It is only mentioned here as an indicator of the value that MBSE and MCE can bring to engineering when we expand the concept based on the evidence demonstrated in our organizational visits with Industry, Government and Academia.

---

## CURRENT STATE OF MODEL CENTRIC ENGINEERING

There is no formalized definition for MCE, which this research has characterized as **an overarching digital approach for integrating different model types with simulations, surrogates, systems, software, hardware and components at different levels of abstraction and fidelity across disciplines throughout the lifecycle, including considerations for manufacturability**. Our organizational discussions found that MCE is in use and adoption is evolving.

We use the term MCE in the most broad way, and see it as an overarching approach to the more general digital engineering because there are no one-size fits all formalized methods, language, or tools; each organization and even different programs inside the same organization implement MCE differently at this time. Critical to our characterization is the capability to achieve multi-domain integrated analysis enabling continuous virtual V&V where the interfaces are formalized: 1) structurally, 2) behaviorally, and 3) temporally in order to use surrogates and simulations in a semantically precise way.

To illustrate the breadth of the applicable topics, we provide examples discussed in our organizational visits. We emphasize methods, models and tools in use. In our data collection, we used a set of factors to collect evidence. If we did not see evidence related to a factor, then that organization did not “get credit” for it, even if we thought they probably did use some aspect of a MCE capability related to that factor.

---

## OPERATIONAL PERSPECTIVE OF MODEL-CENTRIC INTEGRATION

Model-centric views provide a means to integrate different model types with simulations, surrogates, systems and components at different levels of abstraction and fidelity. Figure 2 provides an example documented in a case study that was published in 2008 [57]. The results from the report suggest that the MCE concepts have been applied under various operational scenarios dating back at least eight years. This effort brought together different subject matter experts, who all learned how to communicate using the System Modeling Language (SysML) [93]. Hidden behind the scenes, there was manually created code to integrate the levels and views with simulations, middleware, and legacy components, which evolved into a working system. This reflects on the fact that software skills may be required to assemble model-centric simulations for analysis until we improve the integration and interoperability of models across the domains. However, there has been much advancement in tools, methods and computational power over the past eight years. Our follow-up research [24] [74] and discussions under RT-141 suggests that data interoperability will play a larger roll enabling cross-domain integration, without doing tool-to-tool integration.

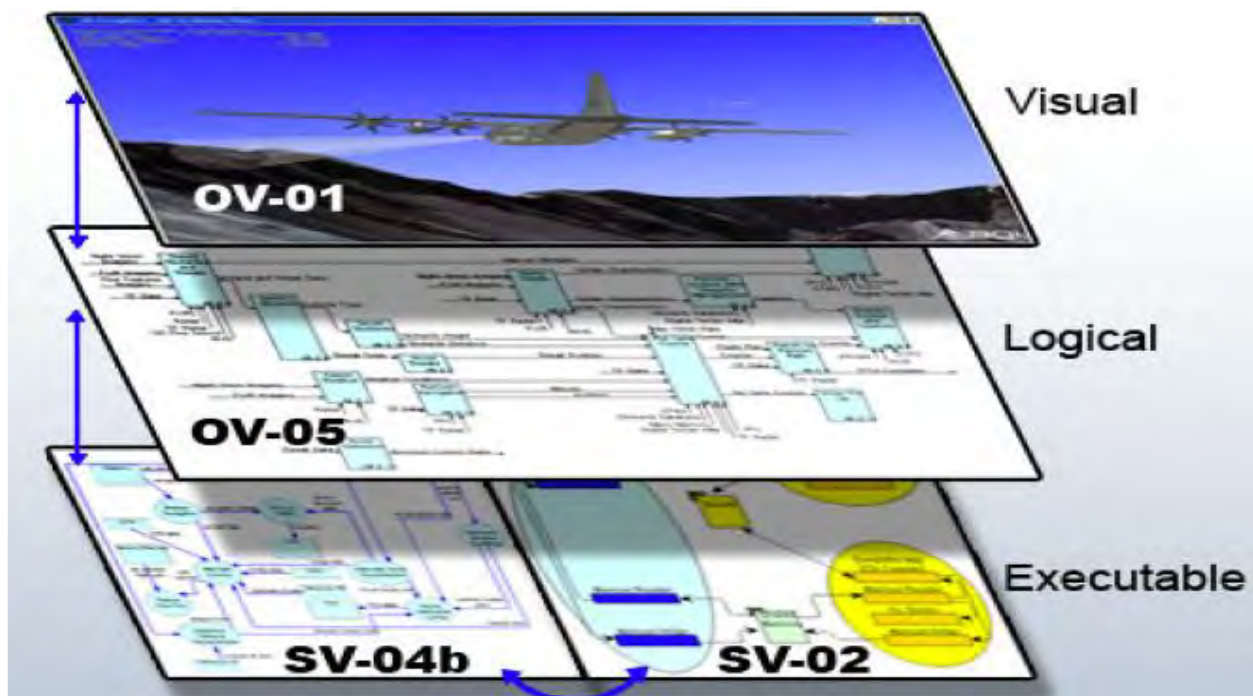


Figure 2. Model Centric Provides Digital Integration Between Views

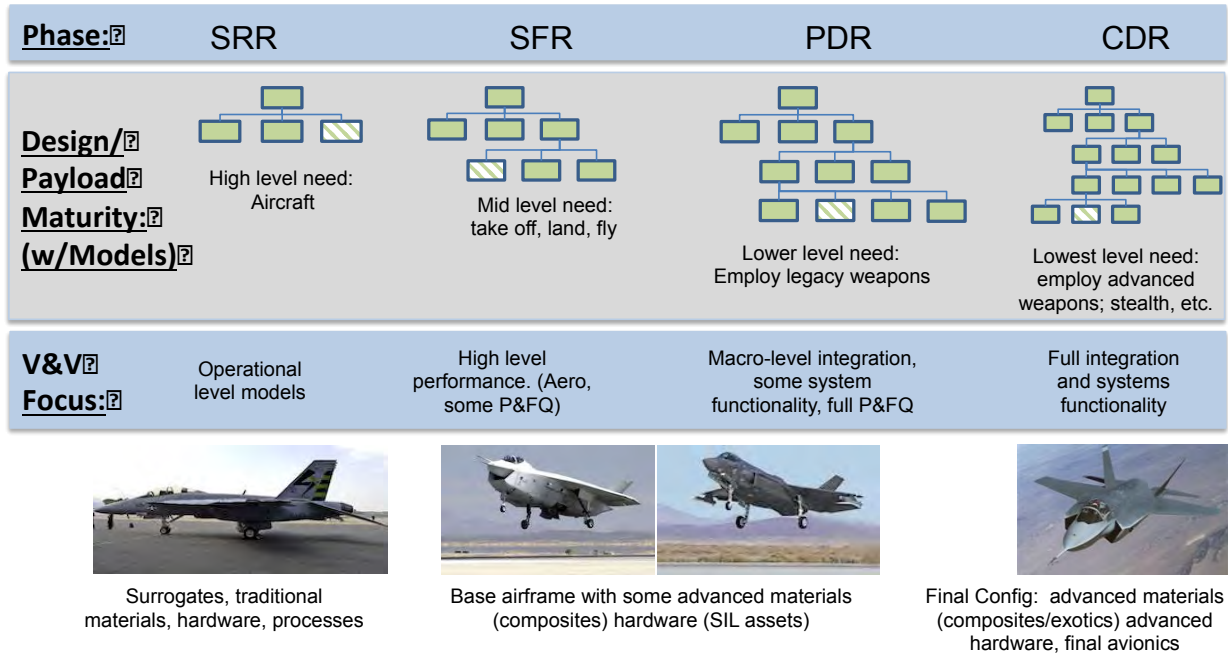
---

#### VIRTUAL VERIFICATION AND VALIDATION CONTINUUM

Extending the previous example and relating it to a scenario of moving through the lifecycle phases, our team provided another representation that was included in the RT-48 final technical report [20] that extends this concept and relates to the concept of iterative and successive refinement across the lifecycle phases. This example is also abstract, but reflects on a NAVAIR objective, which is to continuously “cross the virtual V” early in the lifecycle in order to better ensure that the system design meets the SoS mission needs, as well as identify requirements/design issues earlier in development cycle to reduce cost/impact of corrective actions before coding or “cutting metal,” and ultimately to collapse the timeline. The evidence from the DARPA META projects provides support for this concept [6] [7].

Consider the following scenario using the typical release phase reviews as the time points to represent a notional timeline moving from left to right (e.g., System Requirements Review (SRR), System Functional Review (SFR), Preliminary Design Review (PDR), Critical Design Review (CDR)). In a MCE world the models at SRR would reflect on the new high-level aircraft needs/capabilities, as conceptually rendered in Figure 3; there would likely be a strong relationship between these new operational capabilities and the mission needs. These models would need to be sufficiently “rich” that we could computationally connect them to other surrogates, such as software components (new/legacy), hardware and physical surrogates (e.g., previous generation aircraft). We ideally want to have some type of dynamic operational capability operating from the very beginning of the effort (all digitally). As we transition through the lifecycle phases SRR, SFR, CDR, and PDR, we would use a similar process on lower-level models and components that provide increasing levels of fidelity that is moving us from the analysis of the problem and aircraft needs and closer to the actual system as the decisions for the physical design are defined and refined. In addition, the tradeoffs and impacts of these decisions across the system and SoS are better understood. We begin to focus on detailed functional and timing behavior, with models that predict performance characteristics and begin to clarify the margins and uncertainty; we would continue the transition from

the use of surrogates to versions of the implemented design. As we continue to move through the acquisition phases to CDR, especially for 5th generation air vehicle systems, we will have much more software than ever before, including software that connects models with the simulations, surrogates, components and live or historical environmental data.



\*\*Derived from Ernest S. "Turk" Tavares, Jr. and Larry Smith

**Figure 3. Dynamic Models and Surrogates to Support Continuous “Virtual V&V” Early in the Lifecycle**

Increasingly there will be much more complexity in the software prior to PDR and CDR, and this creates different concerns for NAVAIR from prior generations of air vehicle systems. Testing will be required to ensure that the continuously refined representation of the system models and implementation meet the timing (temporal interactions) and performance needs. This will including testing not only the system itself, but testing required to verify the models and simulation are integrated and working with the system as it is refined over time.

**MISSION-LEVEL SIMULATION INTEGRATION WITH SYSTEM SIMULATION AND DIGITAL ASSETS**

The previous two examples reflect on MCE as it is applied at the system level, which was the scope for our research. However, several organizations discussed mission-level simulation capabilities, including NAVAIR. One organization demonstrated mission-level simulations that are being integrated with system simulation, digital assets and aircraft products providing new types of web-based services. The organization provided a live (with some artificial data) multi-scenario SoS demonstration that runs on a modeling and simulation (M&S) infrastructure that integrates with other M&S capabilities as well as live products that can be hosted within a cockpit or operate through server and web-based services. The scenarios represented a commercial version for a DoD-equivalent mission analyses. The M&S infrastructure is used to both analyze new types of services that can be added to their portfolio and can be used to demonstrate to potential customers capabilities using real or artificial data. These capabilities are used in a way that improves their own systems and capabilities, but they use these capabilities to solicit inputs from potential customers on new types of products and services. However, even with the

advancements this organization discussed, there are some challenges with developing the integrations as there was not a grand architectural scheme or plan when they first started developing the underlying infrastructure. We emphasize this point, because organizations need to apply system engineering principles to their MCE engineering environment.

---

### 3D ENVIRONMENTS AND VISUALIZATION

Several organizations demonstrated (or showed results from) some of their 3D and visualization capabilities [49]. One organization discussed and demonstrated the use of two different types of 3D environments for both customer engagements and for on-going (often daily) design engineering analysis and review sessions in 3D environments. The organization does use commercial technologies but have developed a significant amount of infrastructure on their own. Several similar stories were recorded from others about the need to develop unique infrastructure. A visit was made to the Cave Automated Virtual Environment (CAVE), as shown in Figure 4, where we were immersed in a virtual 3D environment that is used for both analysis and design review purposes.



Figure 4. Cave Automated Virtual Environment (CAVE)<sup>3</sup>

---

### DYNAMIC OPERATIONAL VIEWS FOR MISSION AND SYSTEM SIMULATION AND ANALYSIS

Extending on the example shown in Figure 2, there are modeling environments and frameworks to create dynamic Operational Views (e.g., an OV1), which help to understand and characterize the Mission Context for the needed System Capabilities, as shown in Figure 5. In traditional DoDAF models, static OV1s are used, but the newer capabilities provide for dynamic operational scenarios that not only allow for analysis, but also are being leveraged as scenarios for testing. In many instances these types of capabilities have integrations with other types of models, simulation and analysis capabilities. The current state shows that MCE is moving beyond static DoDAF views. The computational and visualization technologies bring the behavioral views into perspective, but can increasingly bring the temporal aspects into play. This is a capability that can be used at the mission level as well as the system level. Continuing on the

---

<sup>3</sup> Image credit: image credit: media.gm.com



theme of integration, these capabilities are packaged in such a way to integrate with other system engineering and multi-physics simulation capabilities [115].



Figure 5. Dynamic OV1 with Integrations to Other Models and Digital Assets<sup>4</sup>

---

#### MULTIDISCIPLINARY DESIGN, ANALYSIS AND OPTIMIZATION (AKA TRADESPACE ANALYSIS)

One organization gave a demonstration in a Multidisciplinary Design, Analysis and Optimization (MDAO) Laboratory. This organization (call them ABC) mentioned that several years back they had a consulting organization assess their state of the practices against other Industry contractors and it was believed that ABC was trailing the others in the use of MDAO capabilities. They decided that they did not need to do a Return on Investment analysis and just moved forward with putting their lab together. The information they presented showed that they have a much more comprehensive approach today; this includes both integration of the tools, and the methodological approaches. They established the information technology infrastructure to facilitate the integrations across the design space of many facets Aerodynamics, Mass Properties, Performance, Propulsion, Operational Analysis, Ops-support, Manufacturing, and assembly and lifecycle costs across multi-mission scenarios, but not necessarily cross-domain at the same time. They are systematic about creation of design of experiments. They stated that in the use of these technologies, they often uncover or expose things that are not intuitive – that is the more comprehensive analysis allows for many more excursions and they can uncover issues early. The power from the automation and efficiency of the tools often allows them to spend more time doing more in-depth analysis; they stated that they often do 100 times as many excursion of the design space with MDAO versus traditional manual methods.

---

<sup>4</sup> Image credit: AGI

Figure 6 provides a comparison of a Legacy approach to Multidisciplinary Design Optimization (MDO) [51], analogous to Multidisciplinary Design Analysis and Optimization (MDAO). The comparison is based on, and shows:

- Setup time is longer (specification) for MDO
- Execution time is shorter for MDO
- Management time is significantly shorter for MDO
- Reasoning time is longer for MDO
- Number of possible iterations is orders of magnitude more in the same amount of total time

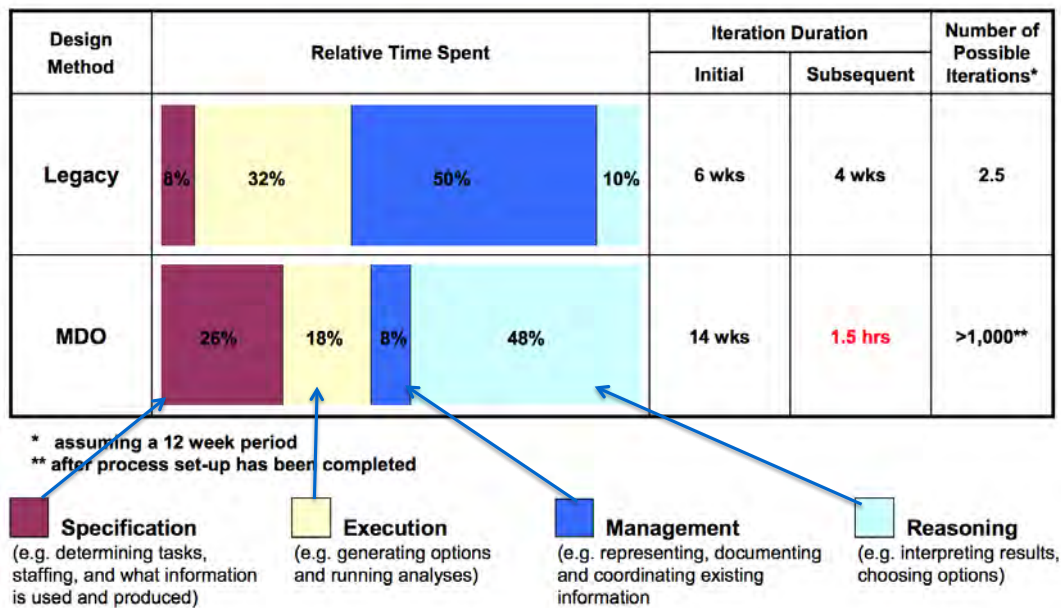


Figure 6. MDAO Compared with Legacy Tradespace Analysis [51]

Once the experimentation framework is setup [51], the case study group was able to do 500 times as many iterations to examine various different alternatives in the same amount of time as with the legacy (manual approach). We heard similar stories during our organizational discussions that were using MDAO. This supports data-to-decisions with engineering technical data and information that is produced, not just documents. There are both commercial and open source MDAO tools and environments that integrate with different modeling tools and many solvers.

#### INTERACTIVE SUCCESSFUL REFINEMENT OF DESIGN SPACE

The DARPA META Advance Vehicle Make (AVM) program also used MDAO for continuous and iterative successive refinement of tradespace alternatives, with considerations for manufacturability leading to “executable requirements” with continuous test at increasing levels of fidelity [7]. AVM Philosophical Underpinnings document, are quoted verbatim below, the objectives were:

*“Raise the level of abstraction such that the designer need not manipulate the design at the lowest numbered part level, but can operate at varying levels of hierarchical abstraction and model fidelity;”*

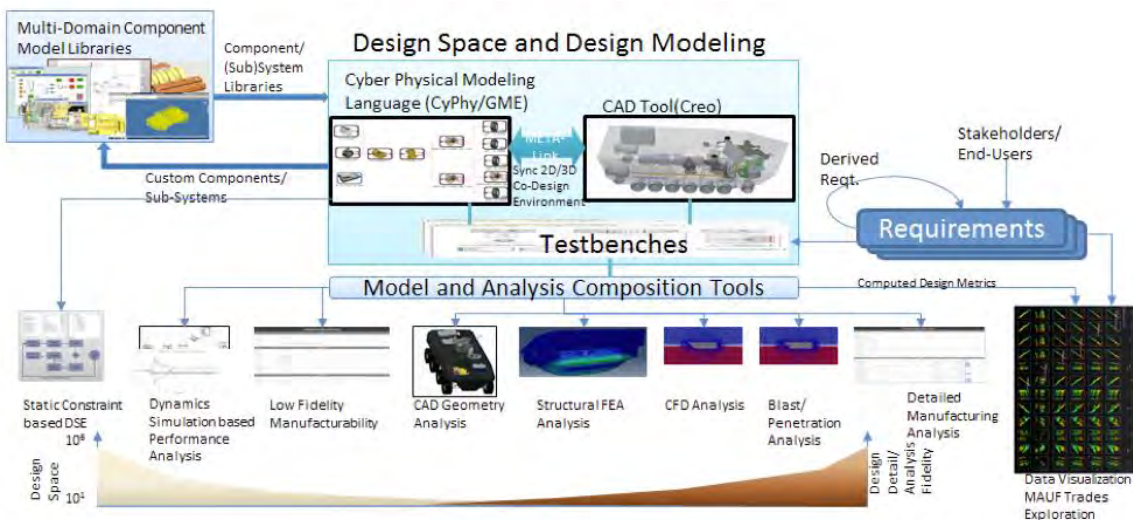
*“Develop practical and observable metrics of complexity to augment size, weight, power, and performance in informing design decisions;”*

*“Enable rapid exploration of the design trade-space for high-fidelity requirements trade-offs;”*

*“Yield detailed system designs that are “correct-by-construction,” i.e., probabilistically verified for consistency, multi-mode interactions, and first-order performance characteristics across all the relevant physics domains (including embedded software).”*

As shown in Figure 7, the overall approach provides evidence for the technical feasibility argument, and another example that relates to our proposed vision concept, namely:

- A more continuous and iterative approach using successive refinement of tradespace alternatives spanning many model types across multiple disciplines
- With data visualization
- With considerations for manufacturability
- Leading to “executable requirements”
- With continuous test at increasing levels of fidelity



**Figure 7. DARPA META Concept**

There were examples of extreme acceleration through HPC where the software allows for doing different types (four-levels) of blast analysis (for tanks); reduced time from months to 3-days of computing time, and a few lessons learned.

## **1D, 2D, & 3D MODEL CREATION, SIMULATION, ANALYSIS FOR PHYSICS-BASED DESIGN**

Discussion with many organizations on 1D, 2D and 3D model creation, simulation, analysis, and management capabilities focused primarily on physics-based design with increasing support for cross-domain analysis. Some organizations have unique capabilities and there is an increasing trend to support broader cross-domain analysis through better integration of different types of models. Some models allow for the plug-in of niche-capability libraries and solvers, using a platform-based approach to create more of an ecosystem (i.e., “think apps”). While yet other models are customizable to leverage High Performance Computing (HPC); that is, they have been programmed to take advantage of parallel computation. There are challenges, but also advances in model transformation and/or interoperability, and the need for formalized semantics is known. There are also multiple suppliers that often provide a suite of tools that cover different parts of the lifecycle, but are not necessarily integrated.

---

## MODELING AND SIMULATION INTEGRATION WITH EMBEDDED SYSTEM CODE

There were many relevant topics that support the vision of model-centric engineering, including one discussion by an organization that performs modeling and simulation of the flying qualities that integrate directly with the code generated from the Simulink model for the control laws of an actual aircraft. By using the actual control law software that flies the aircraft with the modeling and simulation, this increases the accuracy and confidence in the results. This also acts as a type of validation on the control laws software against flying quality simulation scenarios. Finally, the combined integration of the modeling and simulation with the actual code adds to the arguments about the integrity and trust in the models.

---

## PLATFORM-BASED APPROACHES

Platform-based approaches are used by commercial tool vendors as well as developer of systems. The system developers use virtual integration to help refresh systems and do system upgrades on periodic schedules, which in many cases is business critical to companies. For example, two organizations in the automotive space discussed platform-based approaches that are tactically driven by the periodic cycles demanded for sales roll outs at 12, 18, 24, 30, and 36 month delivery cycles. In the 12 and 18 months cycle they might change feature colors but every interface is exactly the same with no electric changes. At 24 months they may make some minor changes including electrical. At 30 months changes to the types of subsystem, components, for example, Figure 8 is based on approach that uses the standards Modelica and Functional Mockup Interface (FMI), which support co-simulation and cross-domain analysis. Finally at 36 months a complete redesign usually occurs.

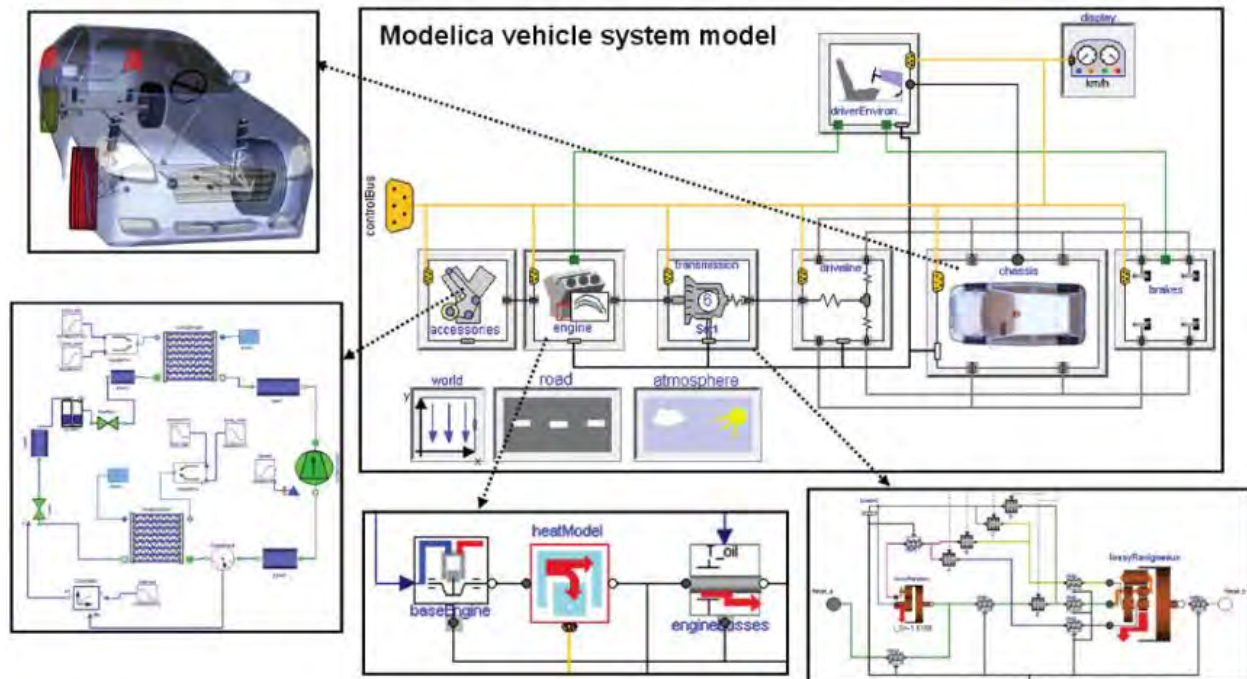


Figure 8. Vehicle System Platform-based Model

There are several implications from these scenarios, and we will list two:

- The automotive industry has been effective at standards-based approaches to platform based development and deliveries, which are critical to cycle times and their business



- They also use their own customized and non-standards based approaches to support platform-based development
- The standards-based approach for integrating models across various domain using the Functional Mockup Interface (FMI), provides a capability as well as a metaphor that would be a way for government to collaborate with contractors across the domains of a system by using an FMI-like backplane to support dynamic analysis and co-simulation of one or more integrated capabilities
  - Functional Mockup Units can be provided in either model-based form, or can be provided in binary form; the binary form would allow contractors to share their dynamic capabilities, but also protect their intellectual property

---

#### **MODELING AND SIMULATION REDUCING PHYSICAL CRASH TESTING**

The automotive companies stated that modeling and simulation is being used to significantly reduce crash testing. Some mentioned reducing cash testing from 320 down to 80 crash tests. This is of particular interest since destructive testing and other types of testing can be expensive and time consuming. The key implication for our sponsor is that simulation may be able to reduce or at least be targeted to specific types of flight tests where there is the greatest uncertainty about performance or margins.

---

#### **MODELING AND SIMULATION OF THE MANUFACTURING**

The DARPA META project [7] and other industry organizations discussed factoring in manufacturability constraints during the MDAO of the design space. Some simulate the manufacturing processes in advance of developing the tooling. One organization discussed model-based manufacturing, model-based inspection, design for manufacturability, additive manufacturing, their smart manufacturing efforts, and advanced design tooling (modeling and simulation infrastructure). In addition, the set-based design concept originally attributed to Toyota described how the design and manufacturing processes work more concurrently. These concepts are strongly related to tradespace analysis and design optimization. This may also provide a means for reducing the time to develop large air vehicle systems.

A rule of thumb is that the effort/time to get from Milestone A to Critical Design Review (CDR) is about 30 percent of the total time, where the time from CDR to Initial Operating Capability (IOC) is about 70 percent of the total time. With some of the new approaches to produce digital information, which considers modeling and simulation analysis of manufacturability prior to CDR, the digital information at CDR could significantly reduce the 70 percent effort from CDR to IOC, which also builds to the argument for being able to reduce the acquisition time by 25 percent with MCE.

---

#### **WORKFLOW AUTOMATION TO SUBSUME PROCESS**

Automated workflows arose from the manufacturing world. A key idea is that if we could formalize all the model-centric artifacts, including the process, we could “generate” information to drive a workflow engine that could significantly subsume the process. This would enable NAVAIR to make decisions in real-time that are completely data-driven. This would also make decision-making more standardized and reduce the dependence of decision quality based on individual experience. The process would be autonomous and adaptive, and coordination would replace planning.

- Workflow automation has the potential to subsume the entire process; everything driven by data, data dependencies; this would be towards a “radical transformation”

- The key reason for this concern/question is that the effort in modeling the “As Is” process is reflecting that it is potentially too difficult to ever fully create or follow a document-driven process
- To a lesser degree, there are other types of products that provide workflow automation support integrations for work such as design optimization
  - We spoke with both commercial companies that provide these capabilities and industry companies that use these technologies
  - They do help speed up and make the design optimizations more efficient, and allow for more iterations, and more systematic regression analysis

A DARPA META project ARRoW (Adaptive, Reflexive, Robust Workflow) developed their concept for a single source of technical truth [6]. This capability allowed them to:

- Automatically generate workflows based on the data/information dependencies
- Integrate model measures that are “automatically” and continuously generated – this relates to a new concept of model measures
- Incorporate knowledge through the use of ontologies and patterns as part of the model reference (reference architecture)

---

#### **SINGLE SOURCE OF TRUTH, MODELING PATTERNS, ONTOLOGIES, PROFILES, CONTINUOUS INTEGRATION**

We heard several organizations discuss the concept of a single source of truth<sup>5</sup>. For example, NASA/JPL provided a perspective on their concept and evolving instantiation of a “Vision” model [8]. They have modeled and are formalizing the overarching Model-based Engineering Environment (MBEE) [43] (designing system) being used to develop instances of a system as well as the mission characterization that is captured in a system model. They continue to formalize the modeling methodology through model patterns [33] [79] that are captured through ontologies, which are associated with a tool-based approach that not only guides development, but provides model analysis to ensure compliance with the patterns (e.g., models are well-formed, consistent, etc.) [69]. This provides their foundation for a single source of truth that is used both for development and continuous reviews.

Among other topics mentioned previously, NASA/JPL has developed an Architecture Framework Tool (AFT) for Architecture Description [9], which provides an overarching perspective on one of the views needed for our Task 3 vision model, and is partially supported with their evolving Open-MBEE [43].

These two concepts are further supported with a rigorous approach to systems engineering (SE); NASA/JPL have identified about 25 modeling patterns applicable to systems engineering. They formalize the patterns in ontologies using Web Ontology Language (OWL) [125] to provide a way of defining a set of concepts and properties applicable to the domain of discourse; in this case not about the space domain, but about the SE domains for concepts such as: component, function, requirement, and work package, data properties like mass and cost, and object properties (relationships) like performs, specifies, and supplies. This provides for a controlled vocabulary and enforcing rules for well-formedness, which permits, among other things, interdisciplinary information integration, and automated analysis and product generation. Because the SE ontologies are expressed in OWL, they are amenable to formal validation (syntactic and semantic) with formal reasoning tools. The approach embedded in SysML and the OWL ontologies is created by transformations from SysML models [69]. Once a model is completed other transformations are performed to the model, such as checking properties of well-formedness and

---

<sup>5</sup> NAVAIR uses Single Source of Technical Truth.

consistency of the model. They currently have about 60,000 test cases for checking these types of properties. The approach is illustrated in several case studies [79].

---

#### QUANTIFICATION OF MARGINS AND UNCERTAINTY

Sandia National Laboratory (Sandia) discussed some of the most advanced approaches for supporting uncertainty quantification (UQ) to enable risk-informed decision-making. The information they provide reflects on the advanced nature of their efforts and continuous evolution through modeling and simulations capabilities that operate on some of the most powerful high performance computing (HPC) resources in the world. We heard about their HPC capabilities, Common Engineering Environment, methodologies on Quantification of Margins and Uncertainty (QMU) [87] and an enabling framework called Dakota [109], which should contribute to our Risk Framework (Task 4). Sandia's team also discussed the various modeling and simulation capabilities and resources that run on the HPC. This ultimately led into a discussion about Model Validation and Simulation Qualification [106].

---

#### GAPS AND CHALLENGES

During our site visits, we asked the organizations to share some of the gaps and critical challenges too, and several of them we have been highlighting in the previous summary. Some provided inputs beyond the question of "technical feasibility," for example:

- Mission **complexity** is growing faster than our ability to manage it
- Not identifying design or **integration** problems until late in lifecycle
  - Complex systems have greater cross-domain dependencies, and many of the modeling and simulation efforts are not doing analysis in terms of the integration of models and their associated simulations
- **Growth and complexity of software** and the verification challenges, which are essential to airworthiness and safety
- **Inability to share** models in a collaborative environment
  - This point again may relate to the underlying semantics of models in specific domains that are not easily shared
- Use of **unvalidated models** in simulations leading to incorrect/invalid results
  - How do we validate models, especially if there is an explosion of models
  - This is part of the model "integrity" need

---

#### CROSS-DOMAIN INTEROPERABILITY

MCE technologies enable more automation and efficiencies; however, there are only a few approaches that deal with cross-domain model interoperability and consistency, and the appropriate use of methodologies is necessary to make these technologies effective. A DARPA META research project cited related challenges, such as the heterogeneity and diversity of domain engineering tools, and the span of design flow activities essential for design of complex CPS; they further noted that the industrial state of the art consists of islands of model-based automation, but supported by tool verticals where integration is limited [7]. One DARPA META project team developed openMETA<sup>6</sup>, an architecture, and CyPhy a new model integrating language to investigate the potential to integrate models and tools in a semantically precise way. Their report suggests successes but also several other challenges. For example a

---

<sup>6</sup> The authors provide examples, but do not promote any particular approach or tooling.

fundamental aspect of their approach involved iterative and successive refinement of a design space through tradespace analysis of assembled component libraries. It was determined that when component libraries were not developed with an appropriate methodological approach that aligned with the capability of the openMETA tools, the approach and system was not as effective as planned. This led to effort to re-model the component libraries in order to demonstrate the new approach and tools. In such a system any set of components often needs to satisfy multiple constraints as it relates to other components ranging from relatively simple structural constraints to fluid dynamics, thermal, software control, reliability, safety, cost, and others.

The openMETA research demonstrated the potential for automated selection of components. However, the lessons learned suggests that the specification of component properties and operational constraints must be aligned with the disparate analysis capabilities of one or more modeling and analysis tools that were not designed to be integrated [7]. Therefore tools alone, even if integrated, are not enough; the appropriate modeling methodologies are also important to leverage the analytical and generative capabilities of MCE tools.

---

### **COMPLEXITY OF SOFTWARE AND THE VERIFICATION CHALLENGE**

The strict requirement for safety and airworthiness for the NAVAIR air vehicle systems requires comprehensive rigor in verification. As 90 percent of the functionality of in a 5<sup>th</sup> generation air vehicle system is in software that implies a significant amount of software verification.

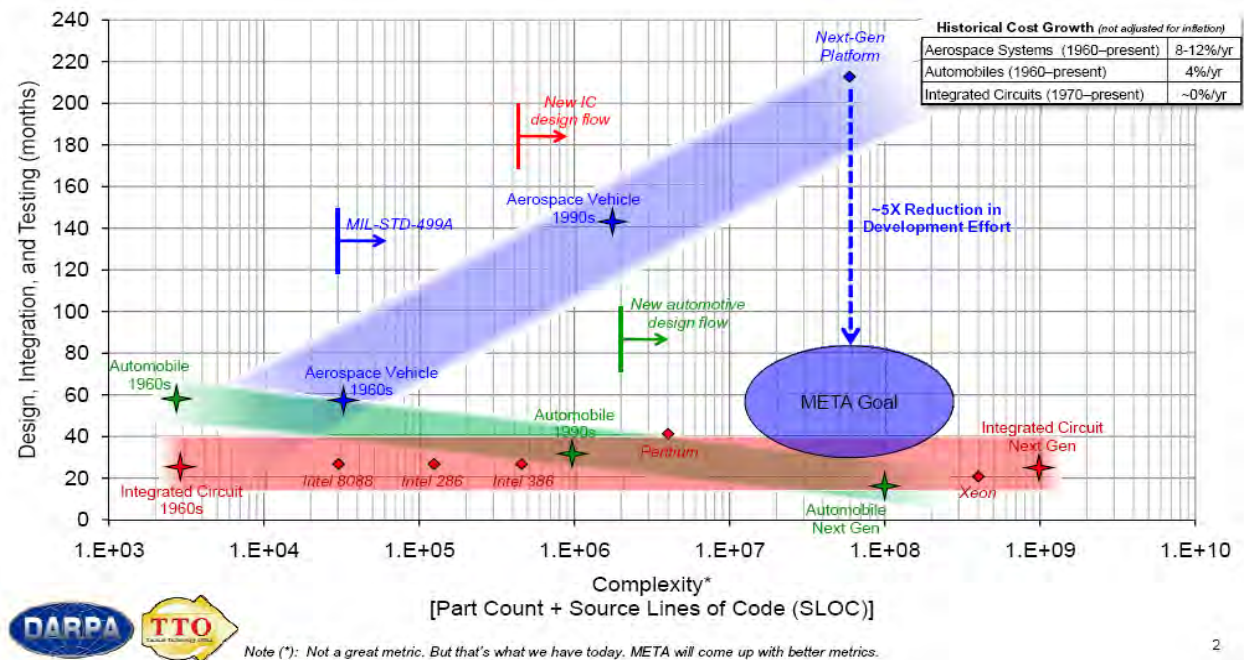
One particular challenge that we discuss in the meetings with organizations is software. Jaime Guerrero, one of our NAVAIR sponsors that attended every organizational meeting, usually discusses his effort on the Joint Strike Fighter (JSF) stating that: “90 percent of the functionality in 5<sup>th</sup> generation air vehicle systems (e.g., F-35) is in software.” There are reports that software testing is taking a long time (GAO report [53]). While there is use of models, the detailed software behavior is often written manually, which minimizes the ability to formalize analysis, generate the code, and automate test, with the possible exception of Simulink (but not everything is modeled like a control system). This is one of the greatest concerns to the goal of reducing 25 percent of the time.

To put this challenge into perspective, NASA presented industry data indicating that verification is 88 percent of the cost to produce DO-178B Level A<sup>7</sup> software, and 75 percent for Level B software [29]. These types of verification requirements are required for many aspects of NAVAIR vehicles, such as the control laws for the F-35. As shown in Figure 9, the DARPA META pre-program solicitation (META) describes how continually increasing complexity impacts the verification costs of software and delivery time [15]. META claims that the fundamental design, integration, and testing approaches have not changed since the 1960s, as shown in Figure 10. The META program goal is to significantly reduce, by approximately a factor of five, the design, integration, manufacturing, and verification level of effort and time for cyber physical systems. The complexity has increased for integrated circuits, as it has for software-intensive systems, but the developers of integrated circuits have maintained a consistent level of effort for the design, integration and testing efforts, as reflected in Figure 9. The need is to understand key reasons why software-intensive systems production is different from integrated circuits. One fundamental difference is that software behavior requires nonlinear operations and constraints that are implemented on computing hardware where operations are performed and results stored in floating point representations. This makes the automated verification problem more challenging than for integrated

---

<sup>7</sup> DO-178B/C is the Software Considerations in Airborne Systems and Equipment Certification document dealing with the safety of software used in certain airborne systems. Level A is a characterization for the most safety-critical aspects of the software, and required a more comprehensive amount of software verification.

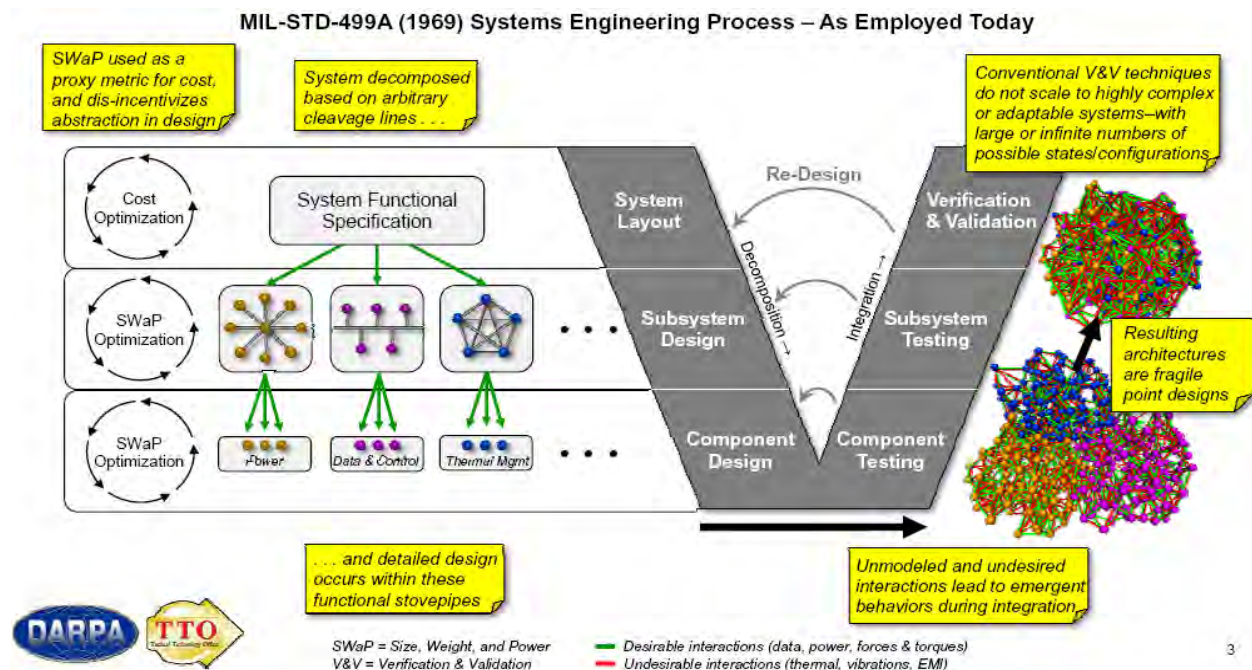
circuits, where automated verification and analysis is based primarily on logic or bit-level manipulations. Chip developers used to rely on simulation, much like software development uses debugging and manual testing, but the chip verification would cost more than 50 percent of the effort and defects that escape to the field could cost \$500M<sup>8</sup>. They now rely more on formal methods and tools to support development and verification.



<sup>8</sup> [http://en.wikipedia.org/wiki/Pentium\\_FDIV\\_bug](http://en.wikipedia.org/wiki/Pentium_FDIV_bug)

<sup>9</sup> DARPA META program APPROVED FOR PUBLIC RELEASE. DISTRIBUTION UNLIMITED





**Figure 10. META Program Claims Conventional V&V Techniques do not Scale to Highly Complex Systems<sup>10</sup>**

There may be many differences between hardware and software, and we briefly summarize the points:

- Software behavior often relies on floating point variables with nonlinear relationships and constraints, which is not the case in integrated circuits
- This requires different mechanisms for analysis and verification than are used in hardware analysis and verification
- Other than models like Simulink, the detailed software behaviors (functions) are still written mostly by hand, limiting automated analyses
  - Some discuss the use of automated generation of code
  - But many are using coding frameworks, which can generate the code structure, but the detailed behavior is written in the code using languages like C++
  - Newer approaches that rely on domain-specific modeling are being researched through DARPA efforts, but most have not become mainstream [22], [99].

Finally, we had discussions with organizations that are researching the use of quantum computing focused on addressing the ever-increasing challenge of verification and validation (V&V) in systems that are increasing in complexity. They stated that V&V costs are growing at the fastest rates of any system component and rates are expected to accelerate with exponential growth in software size and system complexity driving exponential growth in certification costs. These types of technological breakthroughs can also be factored into our scenarios that model-centric engineering will change how we work, and that will reduce or eliminate some challenges.

#### **METRICS AND TOOLS FOR VERIFICATION AND VALIDATION OF CYBER-PHYSICAL SYSTEM**

The National Institute of Standards and Technology (NIST) Foundations for Innovation in Cyber-Physical Systems report [84] as well as the European ARTEMIS Research agenda [4] points out similar needs, as

<sup>10</sup> DARPA META program APPROVED FOR PUBLIC RELEASE. DISTRIBUTION UNLIMITED

stated in the previous paragraph, across many CPS domains. The NIST report identifies 21 barriers and challenges for CPS reliability, safety, and security. In the top rated category of *Metrics and Tools for CPS Verification and Validation (V&V)* the NIST report cites the top three challenges as: 1) the need for increasing coverage of V&V while reducing costs, 2) coping with complexity and scale of systems when performing V&V, and 3) the inability to apply formal methods at appropriate abstraction levels. The hidden challenge again for CPS is that V&V must be comprehensive across the related domains.

History suggests that there is limited success in exposing cross-domain issues until a physical integration is developed to enabling integration and system testing. In the case of systems such as the Joint Strike Fighter (JSF) this has resulted in program delays. This could be due to complexities, leading to late discovery during integration and test related to need capabilities implemented across the domains that cannot be captured using document-centric requirements. There is still research needed, because there is not one clear choice for a method and toolset to support model integration across the domains of a system like JSF, which is needed for design and early virtual V&V of a CPS. This is especially important to acquisition organizations that perform the role of Lead System Integrator.

---

## SUMMARY

We wanted to provide some evidence of traceability to different instances of use of MCE-relevant and cross-domain technologies. We highlight evidence (marked as an X) if we actually had discussions or demonstrations related to this technology as shown in Figure 11. Due to sizing constraints, this matrix includes only a subset of the organizations that were involved in Task 1. We also relate these to their possible impacts/relationships on characteristics:

- Performance
- Integrity
- Affordability
- Risk
- Methodology
- Single Source of Technical Truth

Lastly, we relate them to engineering efforts discussed in the kickoff meeting for this research:

- Prioritization & Tradeoff Analysis
- Concept Engineering
- Architecture & Design Analysis
- Design & Test Reuse & Synthesis
- Active System Characterization
- Human-System Integration

Discussion Topics (not exhaustive)	Instances Where Discussed (not exhaustive)												Characteristics					From Kickoff Briefing						
	NASA/JPL	A	B	C	Altair	GE	Sandia	DARPA/META (V/B)	DARPA/META (BAE)	Model Center	Automotive	CREATE	Performance	Integrity	Affordability	Risk	Methodology	Single Source of Technical Truth	Prioritization & Tradeoff Analysis	Concept Engineering	Architecture & Design Analysis	Design & Test Reuse & Synthesis	Active System Characterization	Human-System Integration
Modeling CONOPS	x															x	x	x						
Modeling Patterns	x								x					x		x	x	x			x	x		
Multi-Physics Modeling and Simulation		x	x	x	x			x	x		x	x	x	x					x	x	x	x	x	
Multi-Discipline/Domain Analysis and Optimization	x	x	x	x	x	x	x	x	x	x			x	x	x			x	x		x	x	x	
Mission-to-System-level Simulation Integration	x	x	x														x		x	x	x	x		x
Affordability Analysis			x				x						x	x	x	x			x		x	x	x	
Quantification of Margins			x				x						x	x	x	x	x		x		x	x	x	
Requirement Generation (from Models)	x		x					x									x	x	x		x	x		
Toolagnostic Digital Representation	x	x			x				x									x	x	x		x	x	x
Model measures (through formal checks)	x		x			x		x	x							x	x	x			x	x		
Modeling and Simulation for Manufacturability			x			x		x					x	x	x	x	x	x	x	x	x	x	x	
Process Automation (workflows)	x				x				x	x								x	x			x		
Iterative/Agile Use of MCE	x	x	x							x							x				x	x		
High Performance Computing	x	x	x		x		x	x			x	x	x	x	x				x	x	x		x	
Platform-based and Surrogates	x	x	x								x									x	x	x	x	
3D Environments and Visualization	x	x	x	x	x	x	x	x			x		x	x					x	x		x	x	x
Immersive Environments		x	x								x									x			x	x
Domain-specific Modeling Languages	x	x	x	x	x	x	x	x	x		x		x	x			x			x	x	x		
Set-based Design		x				x							x	x	x	x			x	x	x			
Model Validation/qualification/trust							x					x		x		x	x		x		x	x		
Modeling Environment and Infrastructure	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Figure 11. Traceability and Scope of Data Collection for Task 1

When all of the relevant technical factors are considered that are critical to a SE radical transformation through MCE or more generally digital engineering, the technical feasibility question seems to place emphasis on three critical items:

1. Cross-domain and multi-physics model integration, and the associated methodologies
2. Technologies to establish and quantify model integrity
3. High Performance Computing (HPC), which enables 1 and 2

Potentially, even more fundamental is the need for a single source of technical truth (SSTT). The SSTT is an enabler for cross-domain interoperability needed for multidisciplinary design, analysis and optimization for problem and design space exploration. The SSTT requires that all information used to assess performance is semantically consistent with MCE technologies and methods used for assuring integrity and the orchestrated workflow is data-driven (not process driven).

Model integrity is defined as verifying our understanding of the models' margins and uncertainty in what they "predict" or in other words when/how do we trust the models. The ability to use logged data to continually calibrate models/simulation is desired. Cross-domain integration of models may also be a way to have greater confidence in simulation models because results of the models should be closer to the real system data.

As another example, the Engineered Resilient System (ERS) research effort builds on some of the themes discussed, showing how such an approach and support for data-driven decision can subsume process, as shown in Figure 12 [63]:

- Single Source of Technical Truth (SSTT)
- Appropriate views and viewpoints for the different stakeholders
- Multidisciplinary Design, Analysis and Optimization (MDAO)
- Workflow Orchestration – by having the data dependencies being known within the SSTT
- Enabled by High Performance Computing





# ERS: *Data-driven* (Not process-driven)



## Appropriate Views for Stakeholders

**Common Core Platforms**



**Needs (...ilities)**

- Manufacturability
- Affordability
- Reliability
- Sustainability
- Usability
- Testability
- Etc.

**Previous Design  
Successes,  
Lessons-learned**



**Portal**

**Framework Interface**

**Enter data once  
Leverage throughout lifecycle**

## Single Source of Technical Truth

**Rapid,  
Reconfigurable  
Systems**

## Enabled by HPC

**HPCMP  
Resources**

## Multidisciplinary Design Analysis and Optimization and Workflow Orchestration



**S&T Resources,  
Research**

Figure 12. Engineering Resilient System Conceptual Representation of Environment [63] - Enhanced<sup>11</sup>

<sup>11</sup> This figure has been enhanced to highlight the notions of: 1) Appropriate Views for Stakeholders, 2) Single Source of Technical Truth, 3) Multidisciplinary Design, Analysis and Optimization, and 4) Enabled by High Performance Computing

## PART II: PREVIOUS WORK MATERIAL

---

The material in the remainder of the document has been extended or refined from the RT-48/118 Final Technical Reports [21] [22]. Additional details related to this material have been documented in the bi-monthly status or working session meeting minutes.

### TASK 1 - ASSESSING THE STATE-OF-THE-ART MBSE

---

This section summarizes some information about the visits and discussions with industry, government, and academia. Prior to each meeting, we sent our coordinator package to an organization coordinator; the package explains the overarching goals of the research task. We often iterate with the organization about the agenda. In coordinating the agenda with our organizing hosts and at the start of each meeting we usually posed the question:

“Tell us about the most advanced holistic uses of model-centric engineering you have seen in use on projects/programs or that you know about”

We also state the question posed by our NAVAIR sponsors:

Do you think it is “technically feasible” for an organization like NAVAIR to have a radical transformation through model-centric engineering (everything digital) and reduce the time by 25 percent for developing and delivering a major 5th generation air vehicle system?

Most of the discussions with industry and commercial organizations were governed by some type of Proprietary Information Agreement (PIA) or Non-disclosure Agreement (NDA). In addition, some of the provided material is marked in a manner that limits our distribution of the material. Due to the need to sign a PIA/NDA, we are being careful about disclosing those organizations in this report. In addition, because we cannot disclose information about commercial or industry organizations, we limit how we discuss the other organizations too, and reference only published and publicly available information.

We have created over 150 pages of narrative from the meeting minutes, which generalize the information we heard during the discussions. NAVAIR wants to share it with our NAVAIR research team, therefore we are including the following notice on meeting minutes that are distributed to our team, per Jaime Guerrero, Director, SEDIC - AIR-4.1, NAVAIR:

DISTRIBUTION STATEMENT D. Distribution authorized to the Department of Defense and U.S. DoD contractors only. Other requests shall be referred to SEDIC Director, AIR-4.1, Naval Air Systems Command, 22347 Cedar Point Rd., Bldg. 2185/Rm. 3A54, Patuxent River, MD 20670 - (301) 342-0075.

These meeting minutes are not part of the official deliverable, but because this report is an official deliverable and will be publically available, we are not going to include any information about the organizations that we met with. Instead, we provide a generalization through the following narrative and discuss the results in the aggregate.

Our team developed a guideline for our collective NAVAIR team to hold discussions in an effort to understand the most state-of-the-art and holistic approaches to model-centric engineering. The objective for our team members was to facilitate conversations through discussions that draw out insights into leading advances in model-centric engineer. We agreed early on with the sponsor that open-ended discussions, as opposed to surveys, would bring out new and more innovative approaches and strategies.

We were particularly interested in demonstrations of actual technical capabilities. We also wanted to understand the critical gaps and limitations too.

## DISCUSSION NARRATIVES AND MEASUREMENT SUMMARY

We created a collection instrument to provide a constructive approach to conduct a discussion with organizations as well as a way to provide some type of quantitative measure associated with using subjective information to rate the “state-of-the-art” of a holistic approach to model-centric engineering. We use a qualitative subjective approach that computes a probabilistic value associated with crosscutting factors associated with the technical Vision for this task.

The collection instrument uses an Excel spreadsheet as the input mechanism to collect factor values about an organization’s use of MBSE. Each row in the spreadsheet represents the subjective information associated with one organization. The latest version of the instrument includes one organizational classifier and 22 factors.

As shown in Figure 13, the model produces two probability distributions, one for the Technical Risk State of the Art (max of 10), and another for the Technical State of the Art (max 100), shown larger in Figure 14. We think these factor values provide a probabilistic value that is related to the technical feasibility questions, and help in reflecting on the factors that are enablers, as well as help identify where gaps exist that must be addressed through risk identification and management. We made some adjustments to the factor weightings based on some of the early discussions we have had with organizations. We also used a non-weighted spreadsheet to illustrate the sensitivities towards what we consider more advanced factors (e.g., metamodel transformation).

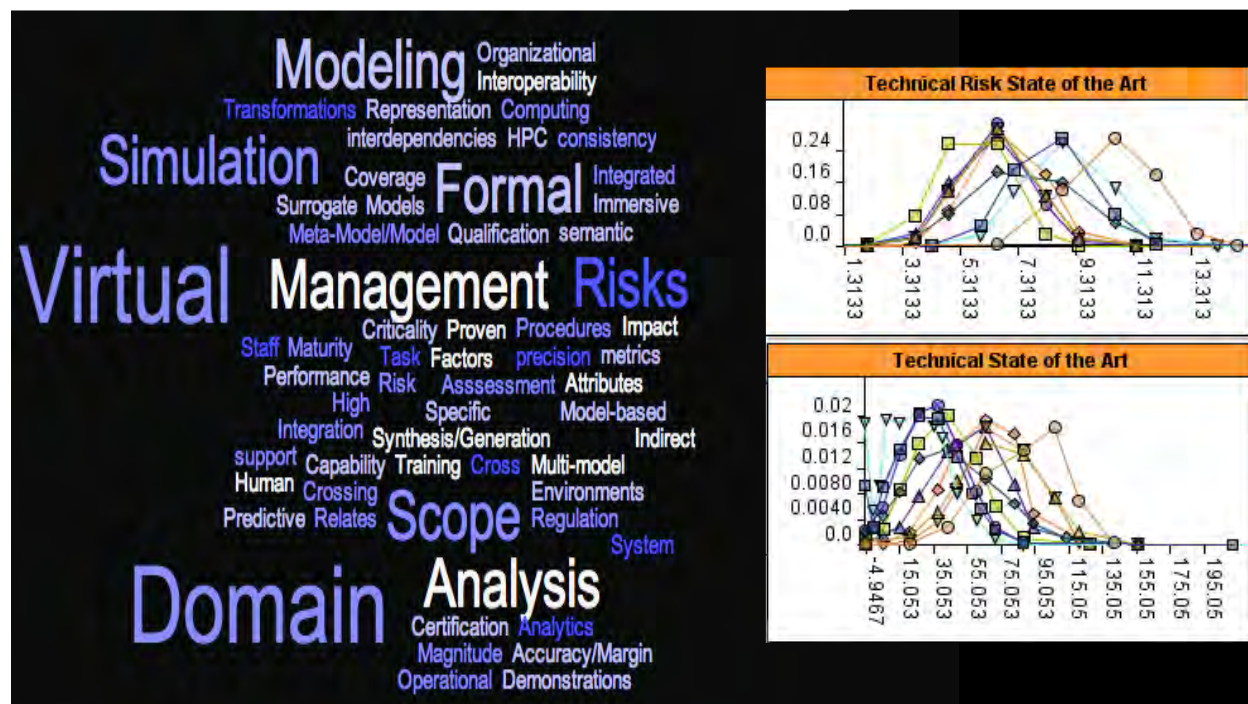


Figure 13. Measurement Collection Instrument

The analysis did highlight several of the challenge areas listed below, but in the end it was not able to deal with the software complexity issue in achieving the goal of a 25 percent reduction in the time to deliver a large scale air vehicle system. Therefore, we addressed this topic with the scenarios provided.

Some of the enablers extracted from our discussions were (this list is not exhaustive):

- Mission-level simulations that are being integrated with system simulation, digital assets & products providing a new world of services
- Leaders are embracing change and adapting to use digital strategies faster than others
- Modeling environments to create dynamic Operational Views (OV1) are more commonly used, which used to be static pictures
- 1D, 2D & 3D Models have Simulation and Analysis Capabilities (mostly physics-based)
- Platform-based approaches with virtual integration help automakers deliver vehicle faster
- Modeling and simulation in the automotive domain is reducing the physical crash testing (e.g., from 320 to 80, and 400 to 40)
  - This could imply that modeling and simulation can reduce test flights, which are very costly as it is difficult to get flight clearances on air craft that have advanced new capabilities
- Design optimization and trade study analysis
- Engineering affordability analysis
- Risk modeling and analysis
- Pattern-based modeling based on ontologies with model transformation and analysis
- Domain-specific modeling languages
- Set-based design
- Modeling and simulation of manufacturing

We next discussed the gaps and challenges:

- Model integration, interoperability, and transformation between domains and disciplines is a challenging issue
  - Still mostly stove-piped
  - Systems engineering is about integration of disciplines across many domains, but there is not a lot of cross-domain integration in the simulation capabilities (only a few exceptions)
  - We have a “sea” of models, simulators, solvers, etc., but we do not have consistent meaning across or between them
  - Lack of **precise semantics** especially in both **behavior** of models and **timing/interactions** of models
  - This limits the full spectrum of analyses and simulations needed to provide adequate coverage over a system’s capabilities; it is also not well integrated “upward” into the mission simulations (although there is effort to do this)
  - Some are looking at how to work and integrate a federation of models and digital assets, but that is not an ideal solution
- Many believe we can “engineer” the “integration” of models/simulations to address this challenge
- Increasing complexity in software, which is 90 percent of the functionality in large scale air vehicle systems
- Use of unvalidated models
  - Note: unvalidated does not mean that the model is invalid

---

## TASK 1 - PROCESS

It was decided by the sponsor that to obtain a uniform perspective on the feedback from the organizational discussions that the Principal Investigator and lead NAVAIR representative for this task would attend all of the organizational meetings. After a meeting with an organization, we:

- Completed one row of the spreadsheet
- Wrote a summary reflecting on the key unique capabilities of the organization
  - Most meetings resulted in five to 10 pages of narrative

The spreadsheet responses are incorporated in a master list. The value for each factor is entered in a modeling tool, which quantifies the subjective inputs provided to the tool, as shown Figure 14. The maximum value of the mean of the probability distribution is 100. As reflected in Figure 14, it was decided that because there are some organizations that require confidentiality or proprietary information agreements, we have decided to keep the names of all organizations anonymous. In addition, the narrative highlights the most key capabilities and challenges, but is generalized to ensure each organization's anonymity. Additional details about interpreting the results are provided in Section 0.

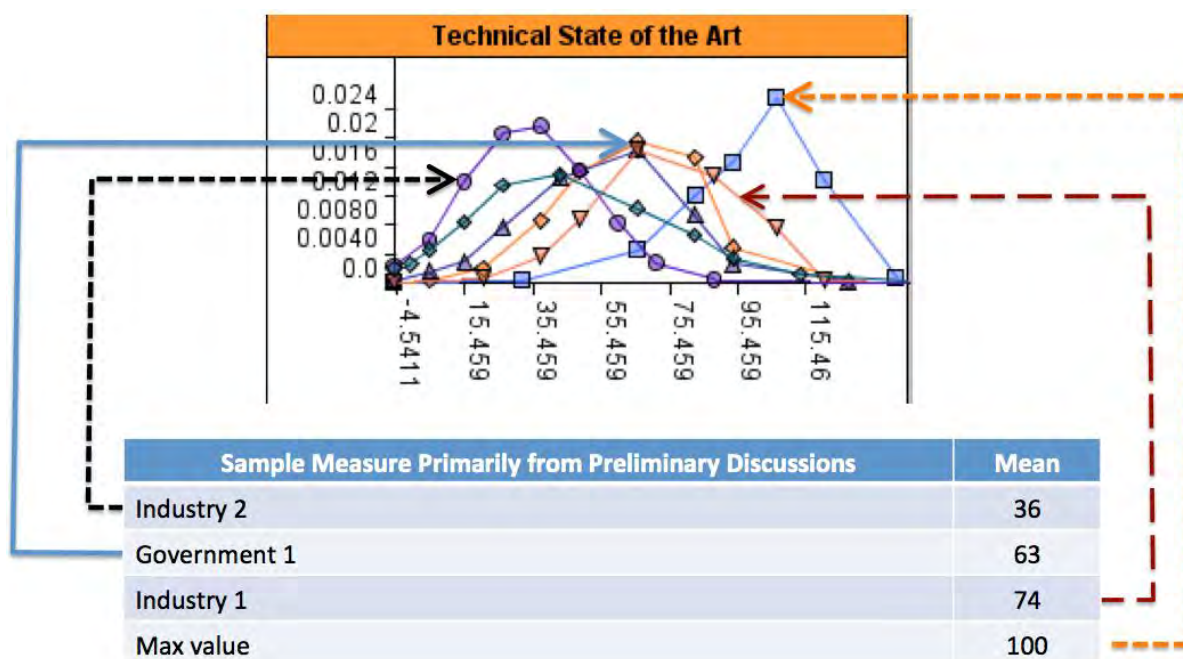


Figure 14. Collection Instrument Results

---

## SCENARIO COLLECTION

After each discussion we complete the spreadsheet collection mechanism as shown in Figure 15 by working through the row and uses the pull down menus to select a factor value of Low, Medium, or High. A complete list of factors is provided in a worksheet tab of the spreadsheet collection mechanism titled: Factor Meaning-Definition. Example definitions are provided in Section 0, with some additional rationale; a complete set of definitions is provided in Discussion Collection Instrument Guide and provided in the back up material of this report. The objective is to identify if there are state-of-the-art methods, tools, processes and innovative strategies that are being used to significantly advance the development and



deployment of systems through model-centric engineering and related approaches, and to incorporate these concepts in the Vision model.

Organization Name

Organizational Scope

Factors (by Category)

Organization (consider the use of an anonymous ID)	Candidate Factors																						
	Magnitude	Proven	Crossing Virtual V	Cross Domain Coverage	Virtual System Representation	Management Criticality Risks (Relates to Task 4)	Attributes of Modeling Maturity	Operational Risks (Relates to Task 4)	Indirect support from Models														
Industry, Academia, Commercial, Government, Other	Organizational Scope	Scope Impact	Demonstrations	Integrated Simulation	Formal Analysis	Domain Specific	Domain Interoperability	Synthesis/Generation Meta-Model/Model Transformations	Surrogate Integration	Formal Capability Assessment	Virtual Accuracy/Margin Analysis	3D Immersive Environments	Risk Management	Predictive Analytics	Model-based metrics	Multi-model interdependencies / consistency and semantic	High Performance Computing (HPC)	Procedures	Staff and Training	Human Factors	Certification	Regulation	Modeling and Simulation Qualification
Industry 1	Site	Med	Pick	Low	Low	High	Pick	High	Med	Low	Low	Low	High	High	High	Low	Low	Low	Low	Low	Med	High	High
Academia 1	Department	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick
Commercial 1	Business Unit	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick
Government 1	Program	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick	Pick

Menu for selecting factors value (L, M, H)

Figure 15. Spreadsheet Instrument Collection

#### ORGANIZATIONAL TYPE

The general convention used is:

- Academia – this should include strictly academic organizations; other organizations performing research should either be Industry, Commercial or Government
- Industry – these are organizations that are using MBSE to develop products and systems (e.g., those contractors to NAVAIR that produce air vehicle systems)
- Commercial – this is a special case of Industry that relates to MBSE product developers
  - These organizations either develop MBSE tools and services, or may apply them with Industry or Government customers
  - These organizations are in the list, because they may have insights into some of the leading or novel uses of the tools, and they are aware of the need to continually advance their own product and services
- Government – this includes military, and other non-military organizations such as Department of Transportation, and the FAA

#### ORGANIZATIONAL SCOPE

One challenge for some of the initial uses of the collection mechanism was to appropriately reflect on the organization scope for which these model-centric engineering usage questions apply. Remembering that the key objective of the research task is to assess the "Technical Feasibility" of MCE. We were pleased that most discussions were held with senior and knowledgeable individuals such as: such Chief Engineer, Chief Technical Offer, Program Manager or some MBSE technical experts in the organization. To carry this

a step further, it might also be important to keep the "systems" perspective in mind, because some of the concepts discussed applied to the hardware level, and advanced approaches to software (e.g., the control laws for the F35 are built in Simulink, with auto code generation, and a significant portion of auto test generation), but these types of approaches may only be emerging in use at the systems level. We wanted to understand how comprehensive the use, and also understand the technical gaps.

Finally, this research is not limited to NAVAIR, however when thinking about NAVAIR systems the scope is often quite large and involves large-scale multi-year programs, where the systems are actually built by one or more contractors.

Therefore, we factor in the organizational scope associated with the MBSE discussion: Program, Project, an entire Business Unit, Platform (e.g., a type of a specific aircraft, tank, automobile), Department, or Site. As a result some of the organizations with lower scores had more limited discussions and did not cover a broad lifecycle perspective on MCE.

---

#### FACTORS DEFINITION EXAMPLE

The factor categories shown in Figure 15 do not necessarily relate to specific MBSE practices, rather they are higher-level characteristics of the organization's ability to leverage the use of models and the associated technologies that enable simulations and accelerate the analyses, design, synthesis, V&V and manufacturing processes. For example:

- **Crossing the Virtual V** is a high-level category that has significant weighting in the model, because our sponsor emphasized this as a critical need and the ability to understand the design capabilities through early V&V activities at the system and mission level (as opposed to the subsystem or component level). This factor category has three main factor characteristics:
  - Simulation of Integration
    - If an organization has simulations of integration or integrated simulations across domains of the system, and especially at the "higher" levels of the "V," this is a likely indicator that such an organization is likely to have the ability to understand simulations of the system within the context of a mission, and there is a better understanding of the integration impacts, because the simulations are integrated or represent integration, including critical temporal aspects in simulation
    - This includes the integration of surrogates, use of instrumented systems, actual system components, new prototypes, and/or in development
    - Other attributes of this type of simulation, would be human-in-the-loop, as well as multi-level mixed fidelity simulations that provide the right abstractions at the right level
  - Formal analysis
    - This means that the analysis is automated, because the models are semantically rich; we are looking for automated analysis, rather than looking at humans performing the analysis
    - Models increasingly have more semantic richness that enable automated-types of analysis
    - Models are increasingly being integrated (see factor category **Cross Domain Coverage**)
  - Domain specific
    - These types of systems involve the integration of many disciplines
    - Models need to provide the relevant abstractions that are related to the domain of the engineer performing the work

- Domain-specific modeling is an emerging type of modeling that often provides the relevant abstractions, with the semantic richness to enable automated analysis, simulation, synthesis (generation) and automated test
- DARPA-sponsored research that demonstrated the capability for continuously evolving Domain Specific Modeling and analyses in 2008 as an emerging capability and theme [39], [99]
- Modeling languages like System Modeling Language (SysML) are general purpose [68] they generally lack the semantic richness needed for formal analysis leveraging for example formal methods of automated V&V [21]; while they may be understood by system engineers, control system engineers would prefer Matlab/Simulink, and other engineers may require other domain-specific models and tools (e.g., computational fluid dynamics, radio-frequency, heat transfer). However, SysML does provide an underlying framework for holding system model information [127], yet the models are not executable even with existing plug-in authoring tools [32].

---

## DISCUSSION SUMMARIES

There are detailed meeting notes that were shared with the NAVAIR research, but they were not generally released as many of the discussions with industry and commercial organizations were governed by some type of Proprietary Information Agreements (PIA) or Non-disclosure Agreements (NDA).

---

## PREDICTIVE MODEL

This section is provided for those interested in more details about the mechanism for converting the subjective factors into a quantitative number. The model is created using a Bayesian Network [97] (BN) tool. There are three basic reasons we selected this approach, BNs:

1. Provide for the translation of subjective information into quantitative probabilities
2. Allows for the use of subjective expert qualitative judgment and captures the casual relationship between subjective factors
3. Create weighting for more advanced capabilities such as meta-model transformation; an organization that can demonstrates/uses such capabilities is very advanced in the use of models as this would support an objective of cross-domain integration

The outputs are also probability distributions, which means that they provide some type of range to provide a comparison between different organizations. The specific numbers are not necessarily as important as our ability to compare different organizations and relate the responses back to advanced uses of MBSE and related enabling technologies. While no organization may have all “High” values, this approach provides a way to look at relative comparison in conjunction with the narratives. Each of the nodes in the BN shown in Figure 16 provides a type of weight called a conditional probability. We have used the team’s judgment to weight the different nodes in a way that would relate to evaluating the key question for this task: is it technically feasible to “do everything with model.” In addition, we will refine the weightings as we proceed through discussions.



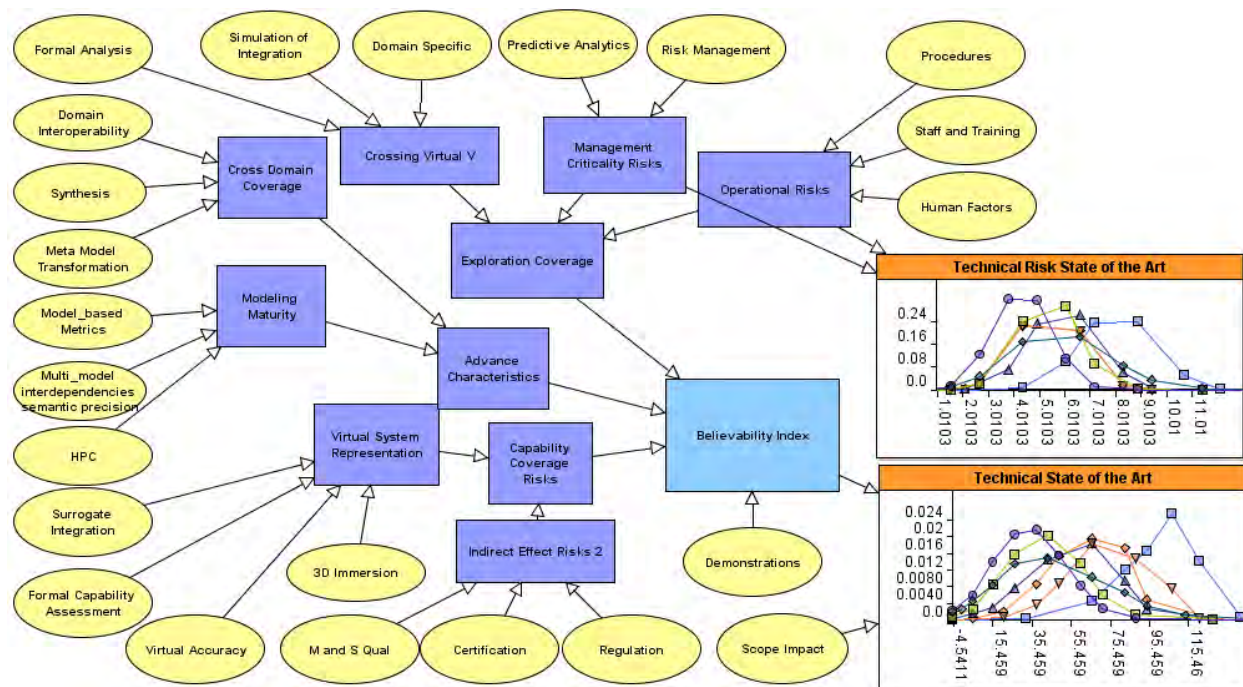


Figure 16. Bayesian Network Underlying Collection Instrument

#### RATIONALE FOR BAYESIAN NETWORKS

A Bayesian network is a representation, which organizes one's knowledge about a particular situation into a coherent whole [40]. They are increasingly being used in the modeling of uncertain and incomplete knowledge. Bayesian thinking is inherently more intuitive than many other evaluation techniques; it best reflects commonsense thinking about uncertainty that humans have. We frequently use words like "likely," "rarely," and "always" to express varying degrees of uncertainty. Subjective probability is our way of assigning numbers (between 0 and 1) to these different degrees of uncertainty, and our probabilities can change as we are presented with new information, or we have new experiences which cause a shift in beliefs or expectations. When this shift occurs, the way our probabilities change are governed by Bayes' rule.

A Bayesian network, as used in this framework, is a joint probability distribution and as such, any question that can be asked in a probabilistic form can be answered with a stated level of confidence. Some typical questions might be:

- Given a set of effects, what are the causes?
- How can an outcome be controlled, given a set of circumstantial values?
- If we model a causal relationship, what result would an intervention or change bring?

While there are several ways to structure a Bayesian network, we used prior experience to structure the model. The subjective factors in the spreadsheet instrument map directly to the yellow oval nodes of the BN model. The purple rectangles are intermediate nodes and generally relate to factor categories. The orange rectangles represent the probability outputs of both Technical state of the art (Task 3) and the Technical Risk state of the art (Task 4).

---

## DATA - LIKERT SCALES (RANKED SCALES)

The subjective factors in the model use a Ranked node type, which is a type of Likert Scale. It is important to note that although Likert scales are arbitrary, they can retain a level of reliability for our use. The value assigned to a Likert item has no objective numerical basis, either in terms of measure theory or scale (from which a distance metric can be determined). In this case, the value assigned to a Likert item has been determined by the researcher constructing the Bayesian network, but can be refined as the research progresses. The results have been a balanced representation of strata and detail.

Typically, Likert items tend to be assigned progressive positive integer values. Likert scales usually range from 2 to 10 – with 5 or 7 being the most common. In this model, 3 levels are used, at least for now as it minimizes the number of computational states, which minimizes time for the analysis. The progressive structure of a Likert scale is such that each successive Likert item is treated as indicating a ‘better’ response than the preceding value. Note that the direction of ‘better’ (i.e., Higher) depends on the wording of the factor definition.

In terms of good practice, a bias in the computations may result if the suppliers of data for the framework do not agree on the relative values of each factor. However, there are enough factors that a bias in one or two values will likely not skew the results significantly.

## TASK 2 - COMMON MODEL LEXICON

---

The team was tasked at the kickoff meeting to create a common lexicon for things related to “models” in the systems engineering domain, and in fact, in the broader engineering space. An example of this is what is meant by the word “model.” This lexicon is focused on providing a common language for all to use in the development and evolution of things related to models.

Most engineers will agree that a model is a facsimile of reality. Yet, to an industrial engineer, a model may represent a production facility; to a mechanical engineer it may be a finite element model analysis; to a systems engineer it may be an IDEF0 [65] or a SysML representation of the system, subsystem, or some lower level element. None of those perspectives are wrong; they are just different views of some part of the same enterprise.

Some claim that there is no existing model lexicon or taxonomy [10], although there are a number of different types of taxonomies that all fit within the more general context of a model lexicon [36], [127]. The Object Management Group (OMG) in conjunction with INCOSE has established an Ontology Action Team to work on similar efforts [90]. The NDIA Modeling & Simulation Committee is about to approve the Final Report on the Identification of Modeling and Simulation Capabilities by Acquisition Life Cycle Phase [11].

Status: we have captured over 750 named lexicon items related to the term “model,” including levels, types, uses, representations, standards, etc. We have had several reviews and it has been refined and restructured several times. The reviewers stated that the graphical representation that was automatically generated from the lexicon was useful.

---

## ONTOLOGY VS. LEXICON

According to Wikipedia, ontologies are the structural frameworks for organizing information and are used in artificial intelligence, the Semantic Web, systems engineering, software engineering, biomedical informatics, library science, enterprise bookmarking, and information architecture as a form of

knowledge representation about the world or some part of it [122]. The creation of domain ontologies is also fundamental to the definition and use of an enterprise architecture framework.

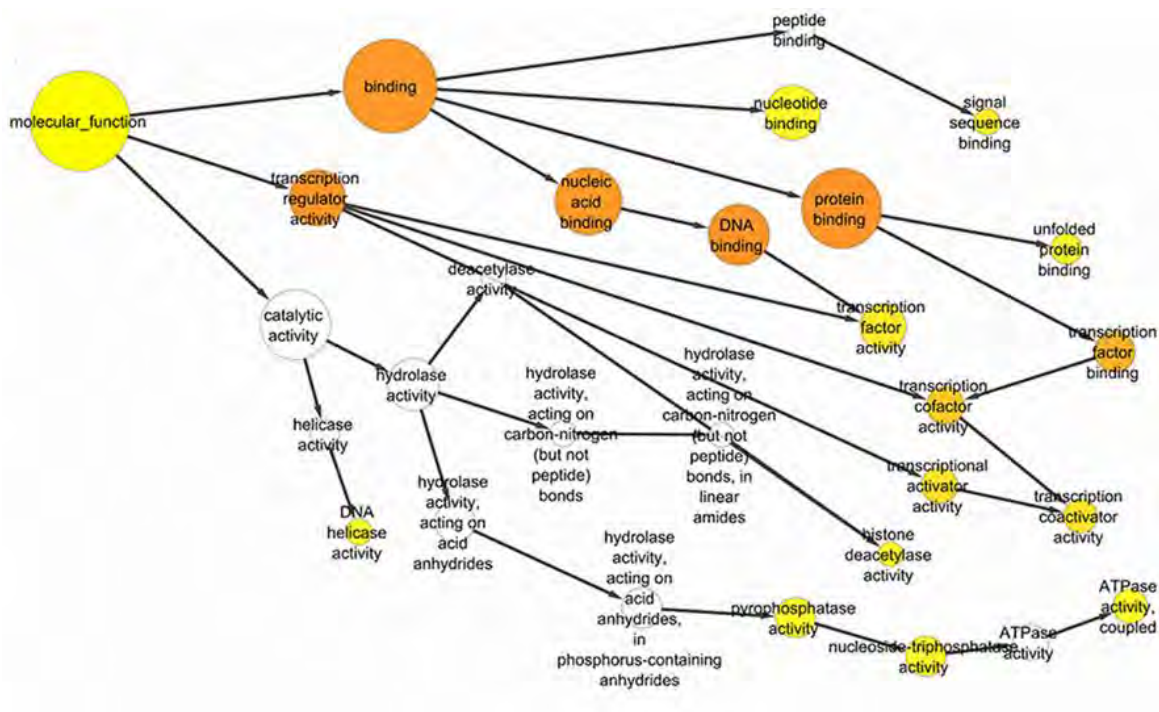
A lexicon is a similar concept – it is normally a book or glossary like document, or words (and their definitions) in a language or domain, arranged in alphabetical order. The team decided that a simple glossary would not be sufficient because it does not show the relationships between terms.

In simplistic terms, an ontology becomes a complex network of words, and their relationships to each other. A lexicon is a glossary. Neither was exactly what was needed for this project. The team needs something that provides definitions and simple relationships to related terms – not complex, rigid definitions. We chose to use the word Lexicon, though the words could also be represented in a tree-like structure that is common for ontologies.

---

## TOOL FOR REPRESENTING WORD RELATIONSHIPS

There are tools available for creating ontologies. There actually exists a class of workers that consider themselves Ontologists. These tools come in many different flavors – from open source tools to commercial tools. The common thread is that they create graphical representations as shown in an example in Figure 17. These tools require rather rigorous definitions and relationships to complete. The open source tools are actually very good, and very robust. However, after some evaluation of available open source tools, the team decided that it would be better to create a straightforward spreadsheet of terms (e.g. a Lexicon), and then create a script that could represent that lexicon graphically.



**Figure 17. Sample Graphic Representation from Ontological Software**

---

## THE LEXICON

A spreadsheet was first created in Excel. At first, the team was simply capturing the words, their definition, and where it made sense, a key reference or two for that definition. Table 1 shows the

implementation of this data gathering spreadsheet. Once the decision was made to create a tool to make this information available graphically, and also on the web, it became apparent that a "relationship" data element was necessary. Therefore, the data collection tool captures:

- Name
- Has Parents [0 or more] separate with ";" if more than one
- Definition
- Sample Usage
- Also Known As
- Key URL (optional)

The current spreadsheet represents a continuous accumulation of relevant terms, their definitions, and their classification. The initial definitions have been drawn from readily available sources on the Internet (often from Wikipedia where the assumption is that it has been created by a group of people with both knowledge and passion about the subject). In other cases members of the research team have authored a definition based on their understanding of the term in a relevant context.

**Table 1. Initial Lexicon Capture Tool**

Name	Parents	Definition	Sample Usage	Also Known As	Key URL or Reference
Metadata	Definition	Searchable information describing the characteristics of data; data or information about data; or descriptive information about an object's data, data activities, systems, and holdings.	Metadata are traditionally found in the card catalogs of libraries. As information has become increasingly digital, metadata are also used to describe digital data using metadata standards specific to a particular discipline. By describing the contents and context of data files, the quality of the original data/files is greatly increased. For example, a webpage may include metadata specifying what language it is written in, what tools were used to create it, and where to go for more on the subject, allowing browsers to automatically improve the experience of users.		DoD M&S Glossary. 2011. <a href="http://www.acqnotes.com/Attachments/DoD%20M&amp;S%20Glossary%201%20Oct%2011.pdf">http://www.acqnotes.com/Attachments/DoD%20M&amp;S%20Glossary%201%20Oct%2011.pdf</a>
Metamodel	Model Types	A model of a model. Metamodels are abstractions of the M&S being developed that use functional decomposition to show relationships, paths of data and algorithms, ordering and interactions between model components and subcomponents.	The Object Management Group (OMG) has developed a metamodeling architecture to define the Unified Modeling Language (UML), called the Meta-Object Facility	surrogate model or response surface models (via <a href="http://sumowiki.intec.ugent.be/Main_Page">http://sumowiki.intec.ugent.be/Main_Page</a> )	DoD M&S Glossary. 2011. <a href="http://www.acqnotes.com/Attachments/DoD%20M&amp;S%20Glossary%201%20Oct%2011.pdf">http://www.acqnotes.com/Attachments/DoD%20M&amp;S%20Glossary%201%20Oct%2011.pdf</a>
Method	Definition	A particular form of procedure for accomplishing or approaching something, especially a systematic or established one.			<a href="https://www.google.com/?gws_rd=ssl&amp;q=define+method">https://www.google.com/?gws_rd=ssl&amp;q=define+method</a>
Metric	Definition	A measure of the extent or degree to which a product possesses and exhibits a certain quality, property, or attribute. (SISO-REF-002-1999)			<a href="http://msco.mil/MSGlossary_TRM_1-M.html">http://msco.mil/MSGlossary_TRM_1-M.html</a>
Ministry of Defence (UK) Architecture Framework (MODAF)	Framework	Ministry of Defense (UK) Architecture Framework			<a href="https://www.gov.uk/mod-architecture-framework">https://www.gov.uk/mod-architecture-framework</a>
Mission	Scenario	is the top-level function of the system; the one that synthesizes all transformation of all inputs and solicitations into outputs and reactions. (Created for SEBoK)	Simulations tends to be broadest in the national security Analysis Department, where multiwarfare analysis requires a variety of such simulations, but mission-level simulations for specific mission areas are also employed in several other departments	Sortie	<a href="http://sebokwiki.org/wiki/Mission_glossary">http://sebokwiki.org/wiki/Mission_glossary</a>

Intuitively, many of the terms in this spreadsheet are ambiguous and their meaning is highly dependent on the context and usage domain. This has been found to be true in reality also as terms are collected from various domains. It is therefore important to emphasize that this is an evolving process.

## SOURCES OF INFORMATION

There were a number of sources used for the initial Lexicon. Journal papers on MBSE provided a good first cut. Interestingly, an article from the Journal of Object Technology [50] proved to be very useful. Other sources included The Open Group, the Object Management Group, INCOSE, NDIA, and Wikipedia. Other contributors include:

- NAVAIR
- OSD(AT&L)/OASD(R&E)/ODASD(SE)/EE
- USAF AFMC AFRL
- Systems Modeling & Simulation Working Group (SMSWG)

- NAFEMS

---

## WEB PRESENTATION

A short script was created that takes the information contained in the data-entry spreadsheet, and generates web-based artifacts. Figure 2 shows the published page as it looks at the time of this report<sup>12</sup>. This page includes four sections:

- Model Lexicon Overview (Figure 18)
- Model Representation/Lexicon (Figure 20)
  - This is a generated image produced by vizGraph, but with over 750 lexicon items
  - For the reviews it was printed out at about 8 feet by 37 feet
- Hyperlinked Tree of the Model Lexicon (Figure 19)
  - As an alternative, a collapsible and expandable tree (outline) allows people to understand the hierarchy of model lexicon with hyperlinks to a particular lexicon definition.
- Definitions - A common structure is used for each term (Figure 21)

---

<sup>12</sup> The final location of the lexicon may move to another location.



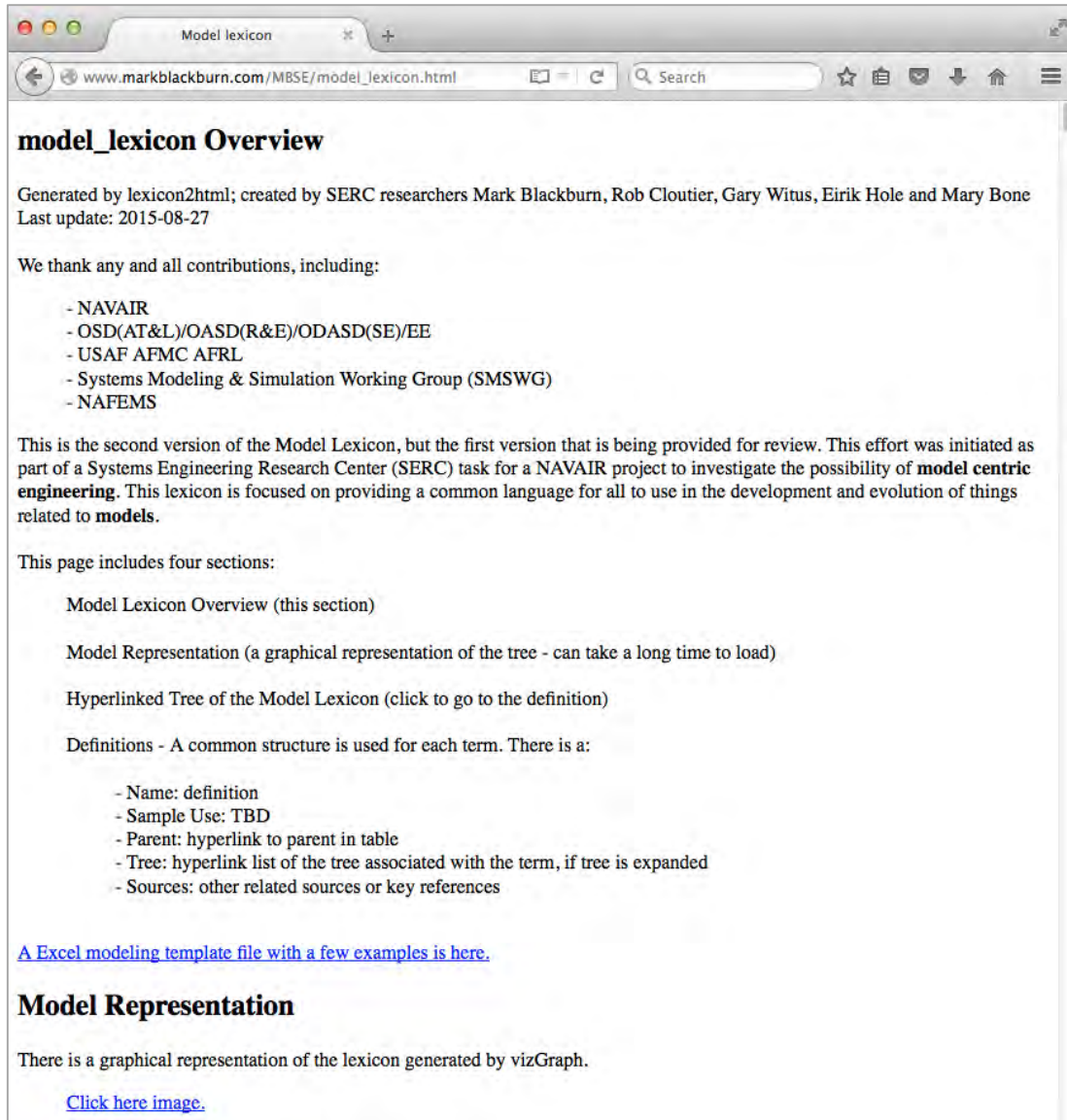


Figure 18. Published Web Page from Data Collection Spreadsheet

## Model Representation

There is a graphical representation of the lexicon generated by vizGraph.

[Click here image.](#)

Note: this is a large image and make take a few seconds to load.

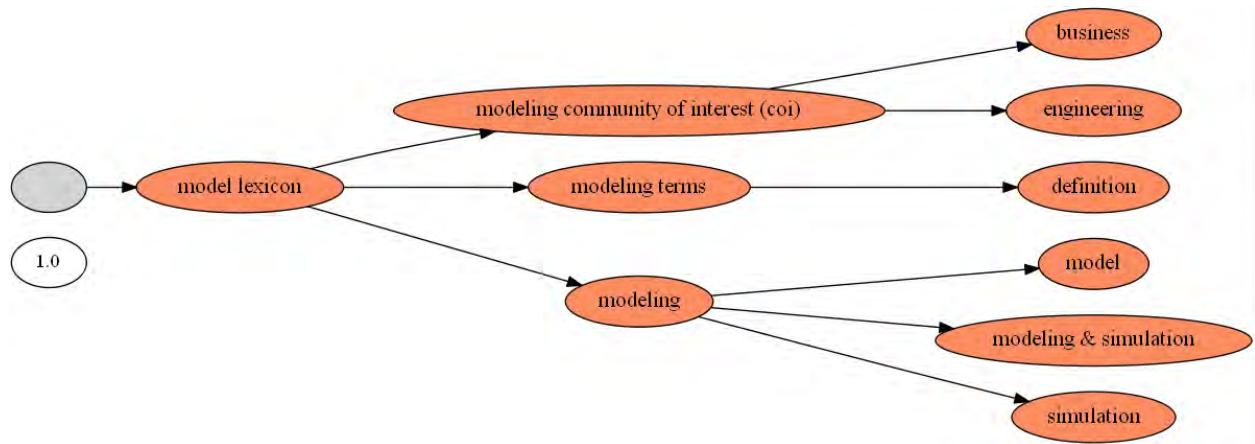
## model\_lexicon

### model\_lexicon Tree

[Model lexicon](#)

- ▼ [Modeling](#)
  - ▼ [Model](#)
    - [Life cycle model management](#)
      - ▶ [Model representations](#)
      - [Model technique](#)
      - [Model transformation](#)
    - ▶ [Model types](#)
    - ▶ [Model uses](#)
    - ▶ [Modeling approach](#)
    - ▶ [Modeling standards](#)
    - [Representation](#)
  - ▶ [Modeling & simulation](#)
  - ▶ [Simulation](#)
- ▼ [Modeling community of interest \(coi\)](#)
  - ▶ [Business](#)
  - ▼ [Engineering](#)
    - [Capability maturity model integration \(cmmi\)](#)
    - [Concurrent engineering](#)
    - [Control systems engineering](#)
    - ▶ [Design](#)
    - ▶ [Requirements engineering](#)
    - [Specialization](#)
    - [Specialty engineering](#)
    - ▶ [Systems engineering](#)
- ▶ [Modeling terms](#)

Figure 19. Model Representation and Lexicon Tree



**Figure 20. Partial Graphical Representation of Lexicon**

The definitions table shown in Figure 21, is a screen image from the website, and includes the following columns:

- Name
- Definition
- Parent
  - This is a hyperlink to the parent in the table
- Tree
  - This is a hyperlink back to the collapsible and expandable tree (outline); clicking on this hyperlink takes the focus back to the name in the tree only if the item is expanded in the tree
- Sample Use
- Key Source (if applicable)



## Model Lexicon Definitions

Name	Definition	Parent	Tree	Sample Use	Source
<b>2d modeling:</b>	a geometric model of an object as two-dimensional figure, usually on the Euclidean or Cartesian plane	<a href="#">mechanical modeling</a>	<a href="#">mechanical modeling, 2d modeling</a>		
<b>3d solid model:</b>	the product of 3D solid modeling	<a href="#">mechanical modeling</a>	<a href="#">mechanical modeling, 3d solid model</a>		
<b>3d solid modeling:</b>	is the process of developing a mathematical representation of any three-dimensional surface of object (either inanimate or living) via specialized software	<a href="#">mechanical modeling</a>	<a href="#">mechanical modeling, 3d solid modeling</a>		
<b>More Sample</b>					
<b>Model driven architecture:</b>	a software design approach for the development of software systems	<a href="#">model technique</a>	<a href="#">model technique, model driven architecture</a>		<a href="http://www.omg.org/mda">http://www.omg.org/mda</a>
<b>Model levels:</b>	Level of the system or system of systems; Also discussed in terms of Resolution level. The amount of detail or degree of aggregation employed in the model or simulation	<a href="#">model lexicon</a>	<a href="#">model lexicon, model levels</a>	Often used in modeling and simulation world to discuss the different types of models	
<b>Model lexicon:</b>	the words used in a language or by a person or group of people		<a href="#">model lexicon</a>	This is a lexicon associated with terms derived from models and modeling	
<b>Model management:</b>	Approaches to managing models	<a href="#">model lexicon</a>	<a href="#">model lexicon, model management</a>	Configuration control, PLM, CATIA	

Figure 21. Tabular Representation of Lexicon

## TASK 3 - MODELING THE VISION AND RELATING TO THE “As Is” AND AIRWORTHINESS PROCESS

Task 3 investigates how to model the “Vision,” “As Is” and Airworthiness process. This was a joint effort with:

- “As Is” process model being worked by Ron Carlson and Paul Montgomery, from Naval Postgraduate School (NPS), and later led by Richard Yates from MITRE
- Airworthiness aspects being worked by Richard Yates and Rick Price from MITRE, and Brian Nolan from SOLUTE
- Vision being led by Mark Blackburn with contributions by our collaborators and many others contributor during our working session

The remainder of this section provides additional details, which is related to the research investigation, working sessions, and task scoping and refinement as it has evolved during this phases of RT-48/118/141. This sections is organized as follows:

- Status of “As Is” and Airworthiness Process Modeling
- “Vision” model context
  - Containing system
  - Designing system
- Operational Perspectives on a Vision Concept
- Scoping the boundaries and interfaces between the Program of Record and Mission Analysis
- System Engineering Technical Review (SETR) Manager
- Modeling and tools for the Vision
- Model-centric engineering perspectives derived from research and discussions

- For example, how model-centric tool chains can subsume process
- Crossing the Virtual V
- Perspective on Integrated MCE Environment

---

## STATUS OF “As Is” AND AIRWORTHINESS PROCESS MODELING

This section provides a brief summary of the “As Is” and Airworthiness process modeling. Historical details on this effort can be found in the RT-48 technical report [19] and RT-118 technical report [22]. The “As Is” and Airworthiness modeling results are being documented in a separate non-SERC report. The MITRE team leads this effort and a final report is expected in December 2015.

The general motivation for defining the “As Is” process and model is to assure the stakeholders that the new process covers everything that was required of the old process. In addition, it provides a type of map to those “As Is” process activities and artifacts that may be replaced or subsumed through new methods and automation.

The Airworthiness process is used to ensure that the necessary evidence is provided in order to get a flight clearance. Brian Nolan from Solute is working with Richard Yates from MITRE to create a model for this process. They have used authoritative sources [70] [74], however, a significant amount of guidance is obtained through interactions with Airworthiness subject matter experts (SME).

There have been a number of attempts and approaches applied to characterize the “As Is” process and Airworthiness constraints. The effort first started by identifying the artifacts that are collected to support the NAVAIR System Engineering Technical Review (SETR) process. The team categorized about 330 artifacts (primarily documents), which were represented in a CORE model [118]. The artifact analysis resulted in a somewhat abstract understanding of the “As Is” process. Many of the artifacts are just named items with no formalized definition of the artifact’s details or resources used to produce them; SMEs knew the key information. This information was needed to understand the information that may need to be captured in a digital engineering approach.

This effort also worked to extract knowledge from stakeholders that have used the processes to further refine both the artifacts and overlay a process. They developed a process model using a standard SysML compliant Activity diagrams representing the actions of three NAVAIR competencies (Software Engineering, Propulsion and Power, and Human Systems) for SETR events, with a focus on how Airworthiness issues are identified and managed. The high-level findings include:

- Current NAVAIR engineering practices are documented at the “What” level but not the “How”
  - NAVAIR practices “good system engineering” and utilizes best-practice and lessons learned
  - “How” NAVAIR applies systems engineering is highly adaptive and fluid and often depends on known/unknown variables
  - A highly prescriptive approach to NAVAIR engineering processes is not desirable and would probably not improve effectiveness
- Airworthiness is not a process but a **set of constraints** applied to the desired capability and the proposed engineering design/solution
  - Airworthiness is one of several constraints that must be considered when developing engineering solutions
  - How and the degree to which constraints are met is managed through a risk management process
  - This is how a balance between meeting requirements and complying with constraints is achieved

Recommendation from the “As Is” process modeling effort:

- How we develop NAVAIR “As-Is” and “To-Be” process models should be adapted to match the nature of NAVAIR work
- Some processes should remain adaptive and fluid to be effective, while others such as Risk could be made more prescriptive and rigorous
- In some cases the focus should be on the result of the process, the data, and not the process – this is inherently what MCE would do
- Knowing who produces what, when and describing the data in a common language may be more valuable than knowing “How”

---

## WHAT IS A MODEL?

We have heard from our stakeholders that some people may not understand what is meant by the term model, as well as having a consistent view on model-centric engineering. We are going to provide some details before moving on to the concepts involved in the Vision.

Modeling, in the broadest sense, is the cost-effective use of something in place of something else for some cognitive purpose. It allows us to use something that is simpler, safer or cheaper than reality instead of reality for some purpose. A model represents reality for the given purpose; the model is an abstraction of reality in the sense that it cannot represent all aspects of reality. This allows us to deal with the world in a simplified manner, avoiding the complexity, danger and irreversibility of reality [107].

George E.P. Box, said: “Essentially, all models are wrong, some models are useful.” [28]

One key aspect of models and modeling is **abstraction**, which supports communication through different views with various levels of details. Details of importance can be emphasized while other details are not described. Most of us have been exposed to models for a long time, for example, a mobile of the solar system, as shown in Figure 22 shows the number of planets and might show the relative position of the planets, but it does not accurately show the planet’s size or distance from the sun. Different views can provide alternative and relevant perspectives on the planets of the solar system and emphasize the relative size of the planets. To get an accurate perspective of a problem or solution often requires several **views** with some type of formal description of the relationship between the views. For example, the distance from the sun to each planet needs to be described using consistent units (e.g., miles).

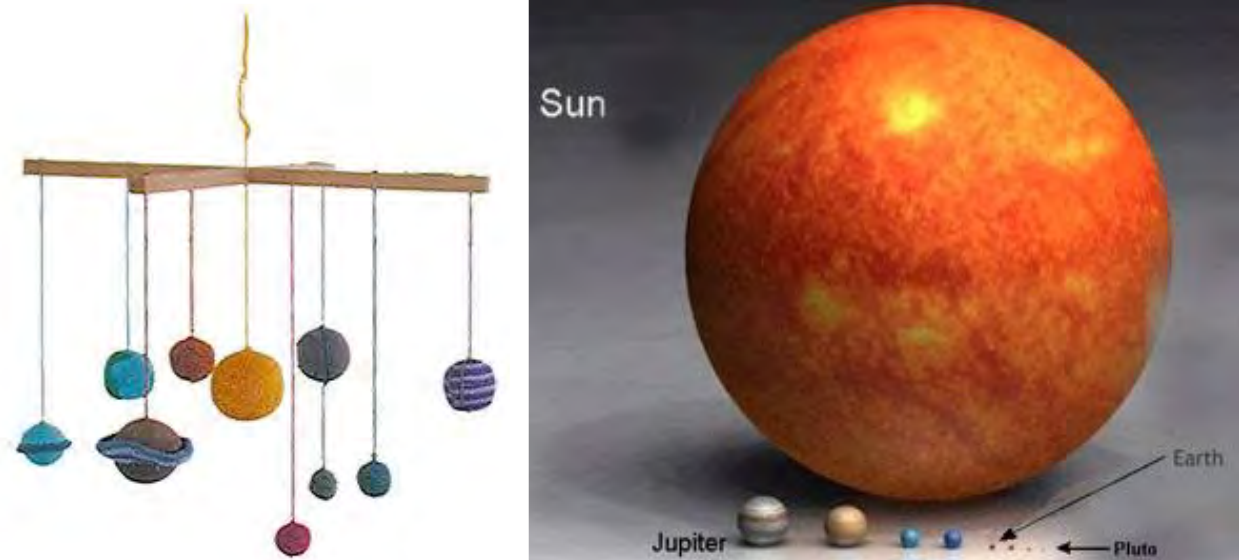
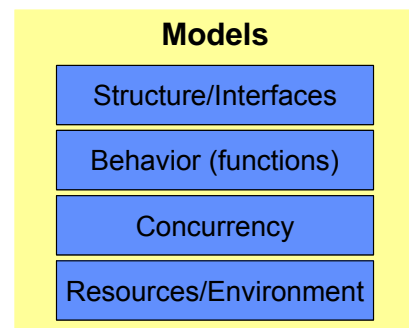


Figure 22. Two Model Views: Mobile and Relative Size of Planets<sup>13</sup>

Model-centric capabilities to achieve NAVAIR's vision will heavily rely on computationally-based models. We need to use the **relevant abstractions** that help people focus on key details of a complex problem or solution combined with **automation** to support the simulation and dynamic analysis of both the problem and solution, along with the mechanism for combining the information collected from the various abstractions to construct a system correctly. Some of the key abstractions can be categorized into types, such as:

- Structure – 1D, 2D, 3D models, systems, subsystems, components, modules, classes, and interfaces (inputs and outputs)
- Behavior (functionality)
- Timing (concurrency, interaction)
- Resources (environment)
- Metamodels (models about models)



Many of these model-centric abstraction concepts have existed and evolved from programming languages, but within a programming language the combination of these views may be lumped or tangled together (e.g., spaghetti code). Most dynamic model capabilities cannot effectively leverage simulation, analysis or generation capabilities if the models are constructed in an ad hoc way.

Modeling methodologies (beyond process steps) are needed to guide the structuring of models to provide a means of systematically separating these views, because certain types of models are constrained to permit only certain types of information. Model-centric automation relies on automated means for analyzing the views, deriving information from one-or-more views, and ultimately pulling sets of views together correctly to produce some type of computationally-based system, simulation or associated analysis artifacts and evidence.

<sup>13</sup> Image credit: [www.thisisauto.com/.../wa07005i/l/ID\\_mobilel.jpg](http://www.thisisauto.com/.../wa07005i/l/ID_mobilel.jpg),  
<http://elronsviefromtheedge.wordpress.com/2006/08/23/and-you-thought-size-mattered/>

---

## VISION MODEL CONTEXT

At the RT-48 kickoff meeting, Dave Cohen presented a Vision concept, and the question was asked “is it technically feasible to model everything?” As a result, we said that we would attempt to model the “Vision.” This term “Vision” was later clarified as a future state<sup>14</sup> with a 10-year time horizon. The Vision aligned with Dave Cohen’s Vision concept as well as the most advanced MCE approaches that would achieve a radical transformation and significantly reduce the time from concept to operations. In the remainder of this section we still use the word “Vision” to characterize the 10-year concept.

Two things have resulted from our efforts in researching what a “Vision” model might be, and how it might be represented:

1. We have seen only a few examples fragments of Vision-like models [8]. Organizations typically model only the systems they want to develop and evolve. Often organizations do not think about modeling all of the elements of the environment, referred to by Ackoff as the **containing system** [1], the elements and interactions of the **designing system**, including elements of existing system, subsystem and parts, in order to create an instance of the target system, which would then be stored in a version of what some call a “System Model” [128] (single source of technical truth)
2. One organization has created at least a start of something that relates to the Vision [8]; they use the System Modeling Language (SysML) [92]. We started an example SysML model to illustrate this concept too. However, in some of our working sessions, we found that not everyone was familiar or comfortable with using SysML modeling **views**.

Therefore, we will provide some examples to help with clarifying the concept of what we think should be included in the Vision, and limit the use of modeling notations.

We typically hear about a System Model [8], [128], which should ideally represent all the data and information to produce the target system, as well as all of the evidence that characterizes the consistency, completeness, and correctness of the information. During RT-141, NAVAIR began to characterize this as the Single Source of Technical Truth. In this case, it is scoped to the Program of Record (POR) for our task. The information to cover what we believe to be the Vision should include:

- Sufficient information about the **containing system** [1]
- This information should come from the mission analysis as sets of desired operational capabilities with performance objectives
- NAVAIR is conducting some similar type of research effort at the mission-level
- All the information about the **designing system**
  - Every tool for model creation, storage, simulation, analysis and their interconnections that is used to create, simulation, or produce analysis information related to decisions or dependent information
  - One organization develops the enterprise, which is the system for producing the target system
    - This is often thought of as the Information Technology (IT) group
    - NASA/JPL integrates and aligns the development of these capabilities with expertise that is used on the program [35]

---

<sup>14</sup> We have discussed this term with our sponsor who also used the words End State, and we stated that there really is no End State as a “Vision” system would continue to evolve as the technologies continue to evolve.

- All other platform related information that provides some aspects of the interrelated capabilities associated with the POR (**system instance to be designed/evolved**), including revisions, variants, and even trade spaces analysis results from design alternatives not selected

Some of these perspectives are provided in Figure 23<sup>15</sup>. This figure puts into perspective the mission capability threads that have relationships to different PORs for the different platforms, and puts into context some of the aspects of the containing system (i.e., the interrelationship to other PORs in order to support a capability). This image abstractly reflects on the information about the existing assets of previous systems (platforms) that can play a role in model-centric engineering:

- The “assets of previous/existing systems” would be represented in some type of reference model or master template<sup>16</sup>
  - All of the existing elements (components) that could be put into a system derived from historical knowledge and reuse
  - The relationships (valid connections) as semantically rich links
  - Model representation of new elements (components) from new design ideas and/or technological advances
  - Figure 8 notional represents this concept for a vehicle
- These types of assets provide the building blocks for defining and refining a new capability
  - For example, existing components (sensors, communication) could be used as surrogate during early evaluation and upgraded as new technology is developed and refined
  - This concept is consistent with what was discussed by the automotive industry and in a more advanced way what was developed by the DARPA META project [7]
- Currently, this information is largely defined in documents; it may be partially modeled, and/or held by contractors (data, models, knowledge)

Therefore, in order to realize the Vision, NAVAIR going forward needs total and continuous access to this type of information in a single source of technical truth.

---

<sup>15</sup> These figures come from a briefing given by Jaime Guerrero that is approved for public release; distribution is unlimited.

<sup>16</sup> A commonly used term is reference architecture. A reference architecture in the field of software architecture or enterprise architecture provides a template solution for an architecture for a particular domain.

Instance: An instance is a specific system that can be developed using the template of the reference architecture.

Any new technology advances should be incorporated back into the reference architecture for reuse in the future.

Variance: Items that meet the same definition in the reference architecture but have different solutions.



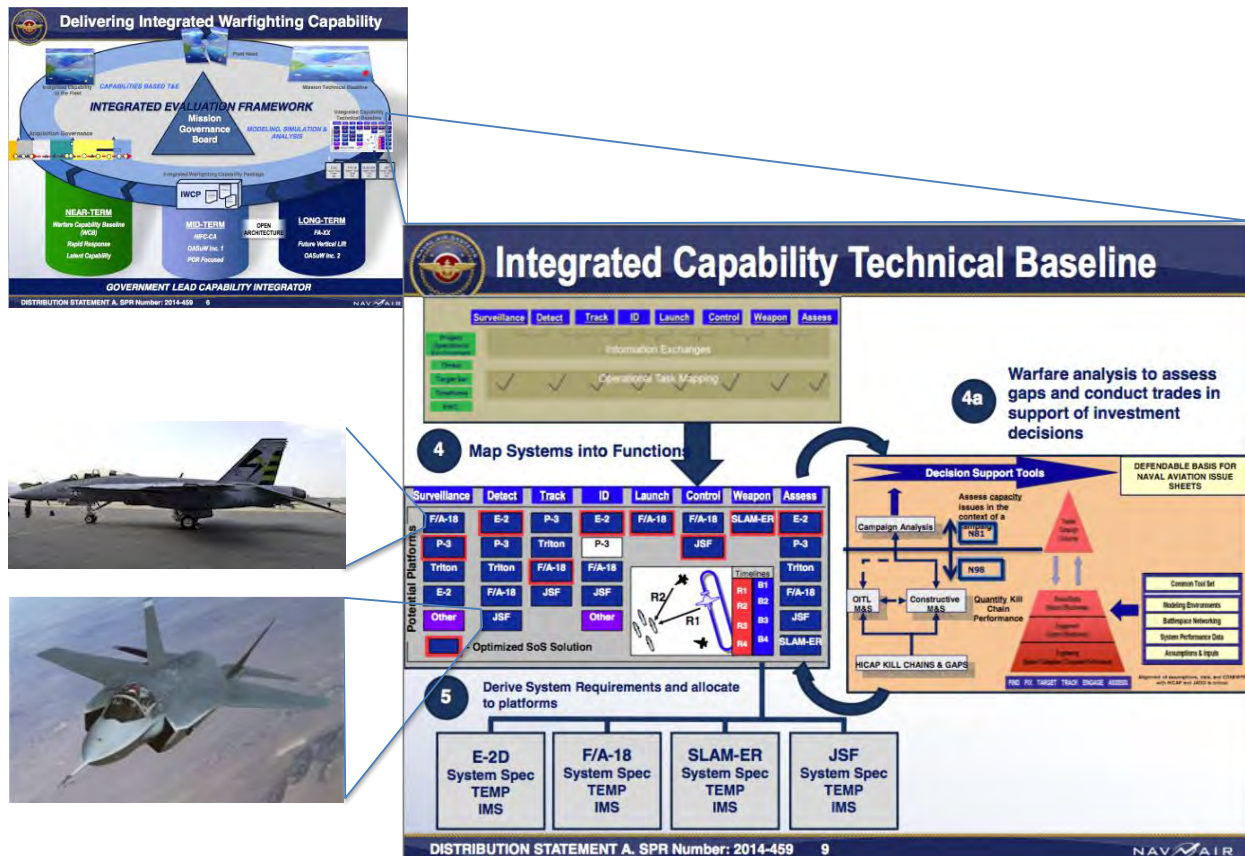


Figure 23. Putting the Vision into Context

## CONTAINING SYSTEM

The Containing System must represent the SoS, including environment and resources with sufficient fidelity and semantic precision to understand how a target system interacts within its environment. These types of views are needed to understand the problem and to investigate, through models, the different alternative systems that can address the problem. In general, a complete high-fidelity representation is not possible, therefore there will be some type of abstraction of the containing system such as reflected in Figure 24. This is one scenario of a capability, and the particulars of the interface parts can include the environment, such as the ship, landing surface, arresting system, weapons, weather, threat types, operators, etc.



**Figure 24. Perspectives on Aspects of the Containing System<sup>17</sup>**

Some of the current approaches to modeling used by NAVAIR such as DoDAF models, are static and do not capture enough information to support simulations. They are semantically imprecise when it comes to representing behavior and the temporal (timing) interaction to be able to assess and predict the needed performance. Our field visits to commercial and industry organizations reflect on modeling and simulations capabilities that are neither well integrated nor interoperable, but some organizations are integrating mission simulations with system products, components or other simulations. While the interest and intent exists, the standards do not keep pace with the technologies.

We acknowledge that there are limitations in cross-domain model interoperability, consistency, and limitations transforming models with the required semantic precision to provide accurate information for decision-making. Without cross-domain integration and interoperability it is difficult to assess the cross-domain impacts, and makes it difficult to understand the uncertainties, which is related to our sponsor's question about model "integrity." We also know that this is an important area of research and there are various approaches that are making progress [7]. This is part of the focus on the path forward.

---

## **DESIGNING SYSTEM**

The Designing System can be the entire enterprise, which includes model capture, synthesis, simulation, analyses, testing, configuration control, workflow automation, product lifecycle management (PLM), model and variant management, etc. The idealized goal is to transform all information so that it can be used for simulation, synthesis, analysis and ideally "models for manufacturing," (e.g., "3D print the system"), models for training, operations and sustainment. This is not realistic for the entire system or all of the parts, at least not today, but this is consistent with the Vision of the sponsor, as well as industry.

---

<sup>17</sup> bigstory.ap.org



We have had a number of discussions with industry suppliers and users of technologies, and there are technologies that support multi-domain and multi-physics-based 1D, 2D, and 3D modeling, analysis and simulations.

The Vision model must include the required information (data) and embedded knowledge that is normally captured in documents, drawing, specifications, pseudo-formal models, and tests (some refer to it as the “total” system model [128]). This concept was discussed in terms of the containing system [1], designing system and ultimately the system instance that is the “design.” This includes or subsumes every piece of information that relates to the artifacts captured in the “As Is” process, but should also include formalized information such as the inputs and outputs of modeling and simulations, analyses, surrogates, risk information, etc. and include specific versions of each tool, simulation, and analysis engine used to support the necessary evidence required to produce an airworthy system version. Ideally, this should include every piece of information to the Bill of Material (BOM).

Discussions with organizations suggest that some individuals and organizations understand the Vision model concept. Some are attempting to develop variants of the concept that are more specific to product development. Some have cross-business/discipline projects established to refine strategies to roll out and support adoption by programs in these different business units. Other efforts are focused more at the software level (using the characterization Model Driven Engineering [MDE]) [64]. One study cited a multi-level, multi-domain instance case that started at the campaign level moving down through the mission, engagement, and engineering levels [3].

There are also organizations that claim to be applying MBSE, yet they have not seen the benefits. We understand that there are often adoption challenges [25]. While this effort was focused on the technical feasibility, the path forward brings in plans for organizational adoption, new competencies and a different business operational model that will address contracting through digital information.

---

## OPERATIONAL PERSPECTIVES ON A VISION CONCEPT

It is a fairly consistent message that many organizations have not defined a Vision model. Instead they are involved in an evolutionary process of model adoption, and many want to better understand the return on investment (ROI). Some organizations have to address airworthiness and safety-related requirements and those efforts can lead to longer delivery schedules. Even the automakers are expending more resources in order to address safety constraints. In addition, some of these organizations are working on a subset of the problem (e.g., V&V) [68], while others are approaching this from the contractor point-of-view, which is significantly different from that of NAVAIR. NAVAIR is working in the early stages of DoD 5000.02 [44] lifecycle (i.e., Milestone A, B, C), and they ultimately produce requirements and design constraints that are provided to the contractors. NAVAIR’s efforts are focused on understanding the problem space in the context of mission analysis.

The objective for the Vision should address the questions:

- Can we create models to cover every type of artifact that is required to produce a system and comply with DoD and NAVAIR processes and requirements (e.g., Airworthiness)?
  - In reality, it is probably not possible, nor practical; therefore, we need to focus on what is valuable [94]
- Can we use model-based simulation, analysis, synthesis and generation to rapidly traverse the “Virtual Vee” and continuously, both horizontally and vertically, leverage evolving digital representations (e.g., models, surrogates) to assess the system design at various levels of fidelity in the context of continuously evolving mission scenarios?

- Notionally rendered in Figure 3 and Figure 23
- How does the risk framework fit into the model?

We initially developed (as a straw man) an example model in System Modeling Language (SysML) that represented the Integrated Warfighter Capability (IWC), which broader than the scope of the POR. The example provided a common understanding that the goal of the modeled Vision is going to formally characterize all of the “data,” relationships, automation throughout the entire lifecycle, including for example the relationship to data used by, and produced by modeling and simulation, analyses and other resources, as well as evidence captured within the models to support risk assessment and management.<sup>18</sup>

We used SysML, because we saw examples from NASA/JPL [8], who is the only organization that we met that has started this type of Vision model concept. SysML works for JPL, because their entire team is deeply versed in SysML. However, we are not sure about the approach for explaining our perspective as we also know that there may be many people in the NAVAIR that are not familiar with SysML. Therefore, we use a storyboard that was created with about 10 different views that should be more generally understandable. We include an integrated overarching perspective that is shown in Figure 25. This image includes information containment and operational perspectives. Notionally starting top down and going clock-wise:

1. This is a **Collaborative Environment**
  - We envision access to this information to be done from at least three forms:
    - Model editor form (raw for the expert modeler, and this could include many types of models, DoDAF, Simulink, SysML, Domain Specific Modeling, Cost model, Computational Fluid Dynamics, Finite Element Models, Reliability, Risk, etc.)
    - Web-based form; view of information synchronized from the “system model;” we have heard many discussion by tool companies, and a similar story about open MBEE from NASA/JPL [43], and this is consistent with technologies discussed by the commercial organizations
    - This would allow for a “dashboard” type web interface, like the SETR Manager that would provide personalized live updates to the user; including prioritizing a user’s workload by allowing them to see how their task affects the bigger program
    - Documents can be automatically generated through personalized or program-standardized templates
  - Access to information is available to all team members and they can see the same instance of information as other team members so this collaborative environment, which provides a single source of truth; security mechanism and role-based view mechanisms also exist today
    - These types of efforts are under way at NAVAIR and more broadly throughout the Navy and other services [117]
2. There is a **Continuous Digital Thread (orange dashed line)** running through all aspects of the concept that is addressing an ever evolving set of needs generically referred to as “**Capability Sets**”
  - Continuous digital thread means that all digital data can be connected and every piece of digital content is aware of other digital content; this is essential for single source of truth
  - The modification of any item can trigger events related to all other dependencies and can change the state of that data, and related data in the single source of technical truth (e.g., trigger weight analysis for entire aircraft if the wing weight increases)

---

<sup>18</sup> There are number of useful representations and documentation that are not currently released for public viewing.

3. **Containing System**, as described in 2.3.1, must represent the SoS, including environment and resources with sufficient fidelity and semantic precision to understand how a target system interacts within its environment.
  - **Capability Sets** are conceptually produced in the context of the containing system through mission-level modeling and simulation analyses to address evolving threats/needs as input from the efforts of the modeling and simulation group
    - We were provided details by two NAVAIR groups involved in mission-level analyses, but will not include that information in this report as it not publically released
  - This is related to discussions at the Mission Level as reflected conceptually in Figure 23 (e.g., operational, and kill chain scenarios, etc.)
4. **Program of Interest** should be an ever evolving instantiation starting from elements in the Reference Model (or Reference Architecture), which are parts of the Designing System
  - We believe that a model-centric approach to a radical transformation will involve the use of “Model Measure” or Model Maturity Levels to assess the state of the models’ completeness, well-formedness, consistency, etc. and its ability to produce all of the needed evidence associated with the Airworthiness constraints
  - During the iterations the capability sets should start converging to a mutually acceptable program of interest
  - New technologies and knowledge captured in the creation of any new system should be captured in the Designing System, including a continual evolution of reference architectures (template of knowledge encapsulation about air vehicle systems and weapons)
5. **Designing System** includes all information it takes to go through analyses and design development; this would include:
  - SETR Manager
  - Every modeling and simulation capabilities, 1D, 2D, 3D, SW, HW, System, Mission, etc.
  - Trade space analyses
  - Reference model (reference architecture) that characterize the architectural structures of air vehicle systems
    - Attributes associated with data about those system/subsystem/components
    - Airworthiness constraints
    - Tools that are used to provide analyses for those different subsystems (e.g., Simulink for control laws)
  - Cost models linked to the reference architectural elements
  - Tools such as Dakota for Quantification of Margins and Uncertainty (QMU)
  - Other risk modeling
  - Cost and schedule modeling and tracking
  - System Integration Labs
  - Surrogates, hardware, software
  - New tools
  - New approach for characterizing modeling maturity measures

This list and story is not exhaustive.

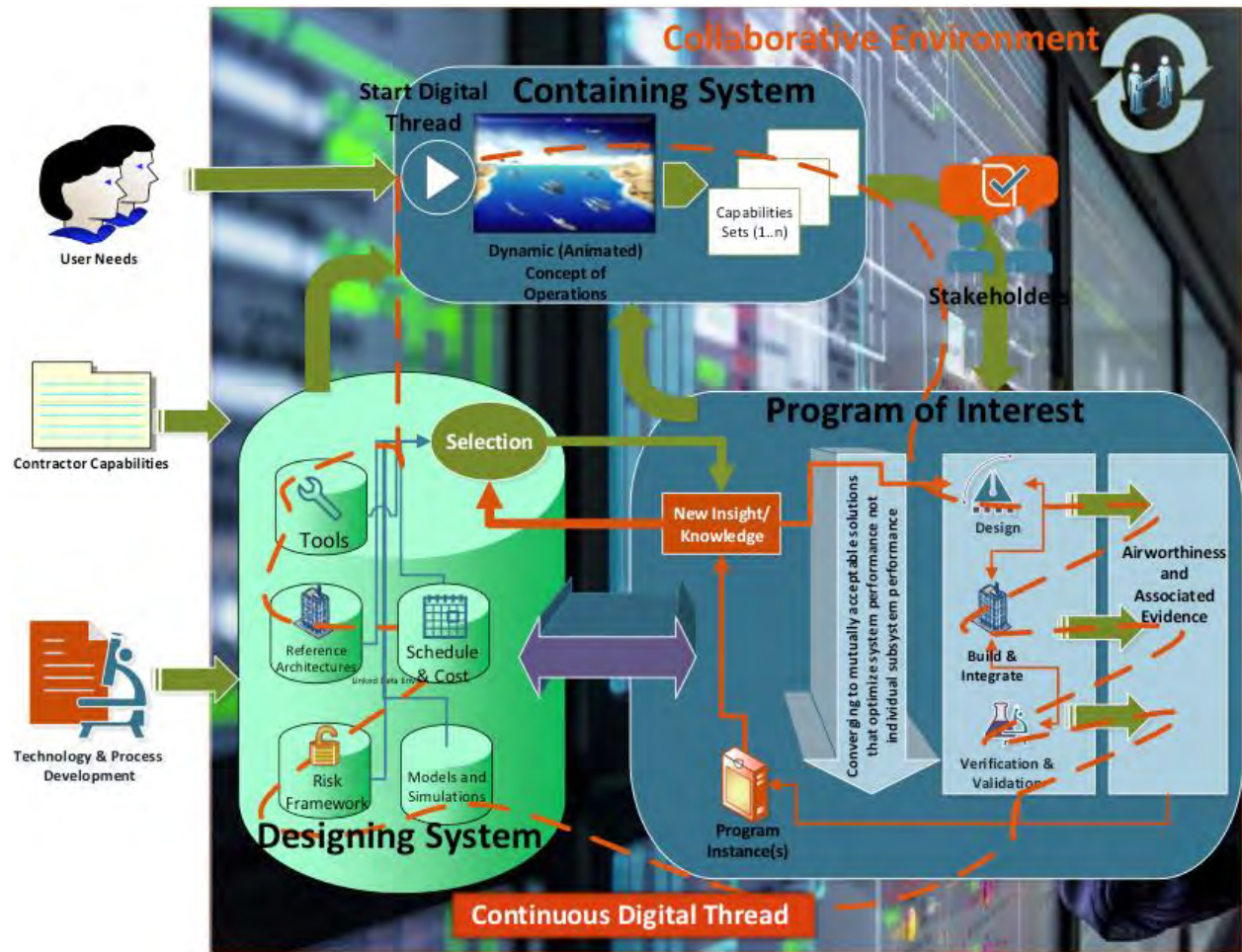


Figure 25. Overarching Concept for Vision

#### SCOPE TO PROGRAM OF RECORD THROUGH DIGITAL CRITICAL DESIGN REVIEW

Figure 23 puts the scope of the POR into context, as well as making the context of a POR part of an evolving platform. This too abstractly reflects on the boundaries between the POR and the mission level. The scope of this research task has been reduced to focus on the lifecycle phases up to critical design review (CDR), for the “As Is,” Airworthiness, Vision model and risk framework for a POR. It was thought that the technical reviews are good “checkpoints” since they focus on different decisions and levels of engineering content that would need to be represented in the models. Only the PDR and CDR are always required. Other reviews such as: ASR, SRR, SFR, TRR, SVR, PRR may or may not be required on a given program. Ideally, we are looking for a new concept: Digital design from CDR artifacts (DCDR). We want to investigate a more continuous notion of PDR and CDR (or DCRD) where reviews of models and analysis artifacts occur “everyday” until all of the required evidence is provided in order to support contracts and signoffs; any meeting can be virtual and in real-time when data is available. This concept is now part of the new SETR Manager.

More importantly, now that we have evidence about some aspects of the technical feasibility question, we want to understand if there are alternative types of model measure that can be used to supplement or eliminate these traditional document-centric reviews as part of the radical transformation. Part of the ongoing research is to investigate if such a concept is viable.

The context for the POR starts from environmental aspects at the mission-level. For many efforts organizations often start with a DoDAF operational view (OV-1) diagram of the mission-level with systems-of-system (SoS) level interactions; increasingly many are using dynamic OV1s such as those reflected in Figure 5, which aligns better with the model-centric engineering concept. The operational views decompose the mission within the context of the situation, and provide different viewpoints that describe the tasks and activities operational elements, and resource flow exchanges required to conduct operations related to scenarios, as reflected in Figure 24.

### **NAVAIR Mission Level Modeling and Simulation (M&S)**

We had a discussion with two NAVAIR M&S groups who are responsible for analyzing the mission scenarios. The groups do have a vision, but not a model of that vision. They indicated that there will be much more cross-domain integration, but the current capabilities appear not to have much integration. The views from these M&S capabilities (i.e. capability sets in Figure 25) define what we have discussed as the **containing system** part of the Vision model, but currently they are not integrated. For our research task scoped at the POR, this information is on the interface boundary, but there is not much that feeds down today; that is, the majority of the analyses from the M&S groups are focused upwards towards the campaign level, rather than downwards towards the system (aka engagement level).

Model-centric perspectives at the POR level would be potentially useful for this effort, because their M&S capabilities must often create some type of abstraction of the PORs and platforms. This is part of the plan for the future, which is to better understand the interface boundary between the M&S level and the POR level within the context of the Vision.

### **NAVAIR Study Views**

Study views were created to address a number of challenges at the POR level and in creating DoDAF requirements. The study view concept builds on lessons learned from creating early DoDAF models; analyses have uncovered that interoperating at the lowest (data) levels is insufficient for scenarios, and scenarios require behaviors, which is missing at the data level. DoDAF does not accommodate other scenario requirements (e.g., conditions, assumptions) very well, and is insufficient to fully characterize the dynamics needed for analysis.

A mission-level SoS analysis begins with formalization using Study Views, as reflected in Figure 26, which has M&S dynamic views and visualization. Study views provide structure and a common context that acts as a basis for framing and bounding the functional decomposition of DoDAF products. Study views formalize the need and intent, provide a situational context and influencing factors to frame and bound the functions and activities of the mission and scenarios that ultimately lead into corresponding representations of the Mission and System Capabilities (i.e., the capabilities for the POR). These capability representations are further analyzed using modeling and simulation and corresponding analysis capabilities. The outputs of which are then formalized in terms of DoDAF artifacts by the NAVAIR Architecture group. This information forms the analysis boundaries for the System Capabilities information needed to define requirements for the POR.



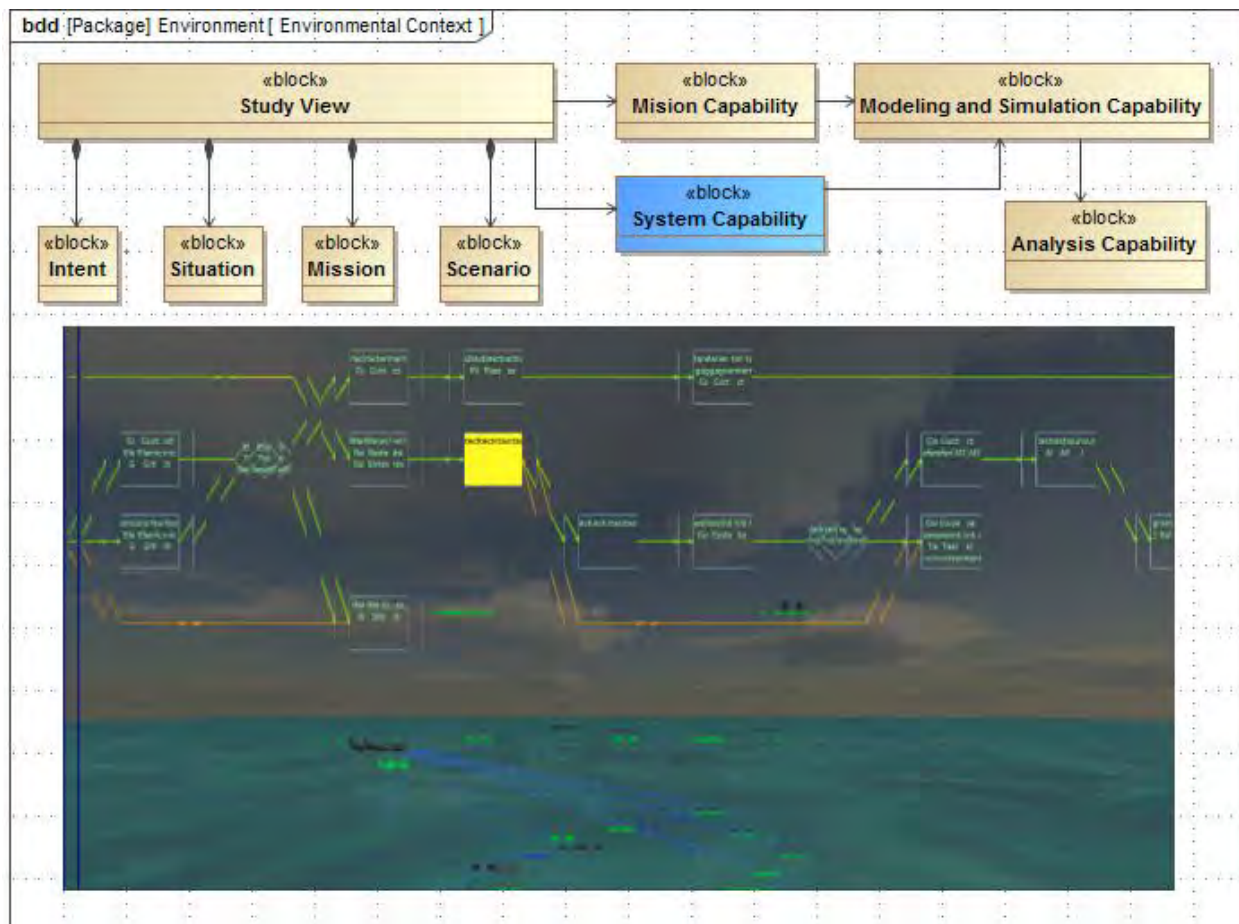


Figure 26. Mission Context for System Capability<sup>19</sup>

## STRUCTURING MISSION-LEVEL ANALYSIS

NASA/JPL, like NAVAIR, has created their own supporting tool that provides for the structured entry and retrieval of architecture artifacts based on an emerging architecture metamodel. We summarize some aspects of it here, because it goes beyond what we currently know about Study Views and is being applied and evolved on Jupiter Europa Orbiter (JEO) project [100].

The architecting focus was elevated to a more prominent and formal role on the JEO project than has been done on most other NASA/JPL projects; the emphasis is to make systems engineering's basic processes, such as: requirements generation, trade studies, risk management, design and interface control, verification and validation, etc., more coherent. The new architecting process used on the JEO project and framework is intended to aid systems engineering in the following ways:

- Adding guiding structure
- Providing better integration of the resulting artifacts
- Ensuring comprehensive attention to important relationships
- Facilitating broad understanding of the architecture
- Maintaining system integrity over the course of development

<sup>19</sup> Image source: Thomas Thompson, Enabling Architecture Interoperability Initiative, B210-001D-0051  
Unclassified.



- Helping to ensure comprehensive verification and validation (V&V)

NASA/JPL acknowledged the choice of a different framework (e.g., not DoDAF, which is used by NAVAIR), because they viewed the choice of framework should be dependent on the nature of the system and circumstances it was designed to support. The JEO most closely aligns with the emerging ANSI/IEEE 1471-2000 standard [67] for software-intensive systems. The architecture artifacts include, but are not limited to, Stakeholders, Concerns, Viewpoints, Views, Analyses, Models, Elements, Scenarios, Properties, and Functions, which align with many of the Study View concerns.

The JEO project team efforts have focused on five objectives:

- Identifying and capturing stakeholders and their concerns
- Developing the content for and capturing viewpoints and views related to the concerns
- Identifying and initiating trades that are needed in the near-term
- Maturing the models that are needed to support those trades
- Training for the growing architecting team

The JEO project team are leveraging the concept and tooling to support a single source of truth<sup>20</sup> [35], using modeling patterns to embed methodological guidance, and computationally applying formal reasoning to ensure that the models are consistent and satisfiable [69], with respect to the ontologies for these various patterns.

### Reference Architecture & Model Based Engineering Environment

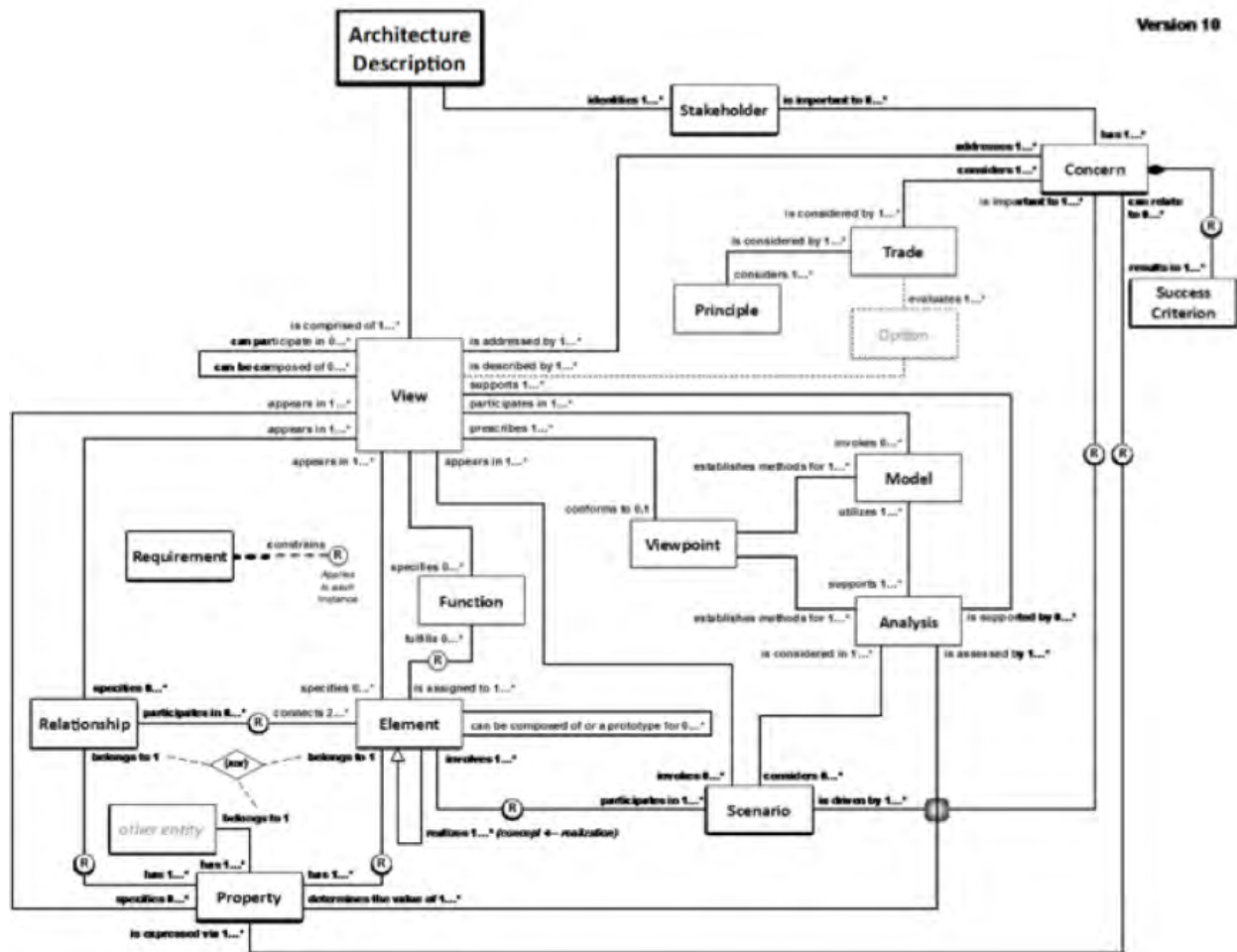
The NASA/JPL projects have a related reference architecture and associated open Model Based Engineering Environment (Open-MBEE) [43] that they are using and evolving on the JEO project. The reference architecture aligns with the vision model concept. They used MagicDraw, which supports SysML/UML [92] and other modeling capabilities to define activities that are transformed to an Oracle database to manage workflow. MagicDraw also provides support to plug-in domain specific modeling tools [79]. They are modeling their artifacts and activities to generate the controls for a workflow engine.

Figure 27 provides an overarching perspective on one of the views extracted from a report [9] that is applicable to the Vision model:

- Blocks in the diagram define categories of items requiring exposition in the architecture description
  - Accompanying each category is a template (not shown) specifying the sorts of information required for each member of that category
- Stakeholders and their Concerns are the drivers for everything else in the architecture, i.e., they can be considered the ‘entrance points’ to explore the framework
  - This is somewhat analogous to the purpose of the Study Views developed at NAVAIR, although NAVAIR does not have a similar representation of its context in a model representation
- The Element is a place holder for aspects of the System to be designed (i.e., Program of Interest in Figure 25)
- The Models, the Analyses performed on them, and the Scenarios, which relate to the “Containing System” (e.g., for a Program of Record) complete the blocks of the Architecture Description

---

<sup>20</sup> NASA uses the term single source of truth, while NAVAIR has adopted the term single source of technical truth.



**Figure 27. NASA/JPL Architecture Framework Tool (AFT) for Architecture Description**

This concept, with the information provided on the reference architecture and the associated modeling patterns provides one of the best stories we have heard as it relates to formalizing the concept of the Vision model. This perspective informed our development of a NAVAIR-oriented concept reflected in Figure 25.

## NAVAIR Architecture Group

The inputs from the M&S group, such as Study Views as shown in Figure 26 are inputs to the System Requirements Analysis and Architecture, which focuses on developing DoDAF views to drive the system analysis and design. They are working toward the requirements for the ~~US~~ Naval Enterprise Architecture Repository (NEAR), which includes the need for Physical Exchange Specification (PES) compliance, however this is a challenge, because some of the tools do not support PES in the same way. While these efforts are using models, they are not using dynamic models. Most important is that these DoDAF type models are increasingly being used in communications with system contractors. While this is not necessarily a radical transformation, it continues to support the concept that sharing information through models is happening today.

---

## MODELING AND TOOLS FOR THE VISION

Our team has had numerous discussions about modeling representations, languages and tools for the Vision. We believe there will be many languages and modeling representations that address both traditional system engineering views such as SysML, but also physics-based and other modeling views. There will also be increasingly much more dynamic information. We will also see an increasing use of Domain Specific Models and associated languages [24] [79] [94], and leverage transformations to support analysis, generation, and alternative types of visualization.

The basic SysML diagrams in the modeling environments are mostly static. System engineering models defined in SysML are descriptive in nature and do not directly produce analytical results [68], nor are they executable [32]. Different tool vendors provide extensions or their own analytical capabilities that solve SysML parametric diagram [26]. Since the parametric relationships are solved as a system of equations, the analytical model is limited to simple equations. To be able to use more sophisticated engineering analyses, external analysis tools need to be connected to SysML models. Other research efforts are attempting to leverage other standard modeling languages such as Modelica [92] (see Figure Figure 8) that have a broad range of analytical support through integration between SysML and Modelica.

Modelica is a standardized general-purpose systems modeling language for analyzing the continuous and discrete time dynamics of complex systems in terms of differential algebraic equations. Domain Specific Modeling environments (e.g., Simulink for control systems) often have richer semantics (e.g., structure, behavioral and sometimes temporal) to support dynamic analyses and simulation; some also have formal method analysis and automated test generation [13] [22] [99]. Other approaches provide process integration and design optimization framework allowing for many available analysis solvers or custom solvers for all type of analysis with simulation and workflow automation [73].

Because SysML is general, there are possible mappings to many types of modeling languages (as is true for UML also) [129] as well as support for programmatic interchange based on the XML Metadata Interchange (XMI) standard [91]. This may rationalize why some organizations are using SysML as an integrating framework, that is, they may not be modeling in SysML, but they are using SysML (and associated tooling) as a mapping or an interchange medium between different modeling languages and environments [8] [43]. While the SysML and UML languages and tools help significantly to formalize the expression, exchange, and graphical representation of system models, SysML and UML languages remain ambiguous and in need of extensions to capture the specific semantics of a given engineering domain [117].

The perspectives cited in this section reflect on why the Vision must go beyond and use other more semantically rich and precise model representations, as well as supporting semantically consistent model (digital) interchange between different simulation and analysis tools. Our efforts planned for the next phase will investigate a potentially more general approach for representing the Vision, and a tool agnostic approach to the single source of technical truth, which we think can support the entire lifecycle [7] [124].

---

## MODEL-CENTRIC ENGINEERING ENVIRONMENT PERSPECTIVES

This section provides additional information related to discussions or actions from our working sessions to reflect on a scenario for how MCE tool chains can provide semantically formalized information between tools to subsume what would normally be manual processes.

---

## MODEL TRANSFORMATION RATHER THAN MODEL EVOLUTION

To reflect on the concept of model transformation rather than model evolution, we provide the following example to describe how model-based automation can completely eliminate manual effort and result in radical transformation of the “As Is” process through an automated workflow [18]. This process was used in the verification of the control laws for the F-35 [88]. This scenario relates to a NAVAIR “As Is” artifact called the “Flight Control Detailed Design Report.” In a model-centric world this type of artifact would:

- Represent “Control Law” in a model
  - Simulink<sup>21</sup> and Stateflow are commonly used to model control laws (e.g., F-16, F-18, F-35) [88]
- Automated analysis that exists today, (e.g., it has been applied to F-35) would include:
  - Satisfiability: proving that each thread through the model has no contradictions (mathematical consistency)
- Simulation
  - Simulation of Simulink models is often done using Matlab
  - Support high-fidelity simulation using Matlab
  - Support higher fidelity with real-time execution within the surrogate or prototype system implementation or actual hardware through automatic code generation
- Synthesis or generation
  - Code generation from Simulink models can be provided by Mathworks and other commercial products
  - Automatic test generation directly from Simulink models [18]
  - Automatic test driver generation
- The test vectors are transformed into various types of test drivers that are run both against a Matlab simulation and the auto-generated code; if all tests pass (the actual output equals the expected output) in both the simulation and generated code execution environments then there is a strong verification argument that the code satisfies the specification
  - Organizations run the test through both the simulation and code, because organizations have been able to find errors in the code generation (Risk reduction argument for using model-based tools)
- Code coverage tools such as LDRA and VectorCAST have been used to show that the tests provide Modified Condition/Decision (MC/DC) coverage [59]
  - Code coverage measurement, which provides quantified risk reduction evidence
- The Mathworks code generation uses a particular algorithm that produces code that is “deadlock” free
  - Eliminates concurrency analysis

These are types of model-based automation that leverage models to “Cross the Virtual V.” While this can be and is commonly done on low-level high-fidelity models, we are also interested in applying this type of concept at the upper-levels of the “V” with varying levels of fidelity that provide integration of model and model automation at different levels of the “V.”

This is a positive story as it relates to the use Simulink-based modeling tool chains that can significantly reduce time by both supporting simulation, code generation, analysis and test generation. However, other forms of software modeling have not had this same type of automation, because behavioral

---

<sup>21</sup> We are not promoting Simulink, we use it as an example, because it is almost a defacto standard for control system modeling and simulation, and it was the tool used in the above scenario.

information in a modeling framework (e.g., UML Rhapsody) is manually coded, and that cannot be analyzed in the same way that Simulink models. This is a concern as software is growing in complexity and size.

---

### **CROSSING THE VIRTUAL “V” BY LEVERAGING MODELS, DIGITAL AND PHYSICAL SURROGATES**

We have continually discussed the notion of “Crossing the Virtual V” as an important way to assess system design concepts in the context of mission scenarios. However, in discussions with organizations, there are some that believe that the notion of a “V” is a historic manifestation of “the” traditional flow of document-driven work, and we should eliminate the use of the “V” as part of systems engineering dogma as it is counterproductive to embracing approaches that support the continuous integration of digital artifacts. The “V” introduces points for disconnects and failure. What is more critical in the Vision is continuous integration of various types of digital assets with varying levels of abstraction and various degrees of fidelity as reflected in Figure 3. Some discussion on this point using examples to further clarify the notion of physical surrogates, and support the argument that the “V” may not be a good metaphor.

The concept of MCE relies heavily on digital assets such as physical surrogates, existing system, or component re-purposed for new concept exploration and prototyping. Our NAVAIR team created a concept for representing System Level Maturity. It reflects on the idea that as we are attempting to “Cross the Virtual V” and will rely on physical surrogates, which is commonly done today, both in aerospace and other domains, such as auto racing. The actual airframe, shown Figure 3 along the bottom matures (right-to-left) and the actual aircraft is first flow (e.g., F-35, 15-December-2006) long before many of the complex software intensive systems are developed and integrated, as the aircraft airframe and new materials are being evaluated. Key early capabilities such as software for the control laws to fly the aircraft are often evolved from earlier aircraft systems (e.g., many versions of MATRIXx and/or Simulink models have been evolved for years, and will continue to be evolved for years). Yet, all of these systems are continually refined and as the timeline of system capabilities mature, new capabilities are added to the system. We believe that in MCE, it will be possible to have continuous integration and tests, much like agile is used in software. Formalized interfaces are required for integration, and the semantics for the interfaces often need to be formalized: 1) structurally, 2) behaviorally, and 3) temporally, in order to use surrogates and simulations. Document-based specifications do not formalize these, however some modeling approaches can, and with semantic formalization, automated verification can be supported directly from the models, which is supported by evidence.

---

### **DECISION FRAMEWORK**

The Army’s TARDEC provided a presentation and demonstration on an evolving a framework called the Integrated System Engineering Framework (ISEF) [52] [117]. Briefly, ISEF is a Web-enabled collaborative decision-making framework to support knowledge creation and capture built on a decision-centric method, with data visualizations, that enables continuous data traceability. The framework integrates a number of different technologies that support decision-making applicable to different phases of the lifecycle, for example:

- Requirements – they have their own requirement management capability
- Feedback mechanism
- Portfolio management
- Architecture (through other MBSE tools)
- Tradespace analysis
- Risk

- Road mapping

While the information from this meeting may not directly address the research question for a radical transformation, the information we received seems valuable, as ISEF is complementary for a transitional model-centric approach. It also illustrates the need for organizations to do some level of architecting even if they integrate other tools. This is likely to be the case for NAVAIR, because they will need to have a more tool agnostic environment.

---

#### **SYSTEMS ENGINEERING TECHNICAL REVIEW (SETR) MANAGER**

This section briefly discusses the new SETR Manager, which is inherently part of the “As Is” process, but we believe it could be part of the “to be” Vision. Similar in some respects to ISEF, the SETR Manager is a server/web-enabled way to navigate through the SETR checklists. It provides real-time status updates and reviews, and allows for discussion tracking providing a familiar Facebook and Twitter style that should provide an easy-to-use look and feel, allowing teams to come up to speed quickly. This capability is a transformation of a few different types of SETR checklist approaches that have structured and layered different types of tooling for the checklist with some reorganization of the checklist questions (more 5000), but layering them. The Tier 4 questions (~1500) are still Yes/No, and the other possible question have now been moved to Tier 5, and are referred to as Considerations, which add context to the Tier 4 questions. There may be a need to move some of the Tier 5 questions to Tier 4.

SETR Manager will be an ongoing evolution, which NAVAIR wants to do in a much more iterative (Agile-way). In its current state the SETR Manager:

- Provides dashboard views of the SETR Manager data for all primary management roles, and competency (tech authority, SETR content owner)
- Uses the dashboard to support drill-down of data
- Visualizes historical trends where possible
- Allows comparisons between different sets of data (i.e. between multiple competencies or programs)
- Steers attention quickly toward potential issues and/or tasks that must be accomplished

The SETR Manager is part of the Designing System, shown in Figure 25, and also believe it to be a key element of the “rich” web interface into the single source of technical truth. The overall metaphor provided by the capability aligns with a much more collaborative way of supporting real-time reviews and consolidated measurements in consistent colorized dashboards, with visualization. The server-based approach allows for an easier and more continuous updates as NAVAIR adapts, and to support integration of other web-enable and server-based approaches for continuous and collaborative engineering.

---

#### **INTEGRATED MCE ENVIRONMENT**

We have provided a number of references and showed images representing overarching MCE concepts that have been created by other organizations, such as Figure 12. There are other notional perspectives created by many organizations, and we use Figure 28 as a notional summary provided in yet another perspective of an integrated environment of capabilities for the vision concept:

- Provides appropriate views for the various stakeholder
- Stakeholders have views into the Single Source of Technical Truth (SSTT)
  - Using rich modeling interfaces for those with expertise in modeling



- Using rich “web” interface, which today provides support for graphics, integrated with structure inputs, generated textual views and 3D model viewing [102]
- MDAO layer provides for problem and design space exploration of
  - Physics-based models
  - Integrity-based models
  - Cost and scheduling models
  - Risk models
  - Various “ilities” models
  - Including surrogates and components
- Enabled by High Performance Computing (HPC)
- Semantically rich linkages between data and information in the SSTT provides for continuous workflow orchestration – enabled by HPC
- Document generation is enabled by
  - Semantically rich links to information in the SSTT
  - Templates that formalize patterns for requirements, contracts, etc.
- Enabling technologies such as machine learning provides a virtual knowledge librarian that assist users guided by embedding knowledge and training
- Contractor and collaborators have a secure means to plugin to view or share digital information as a new paradigm for interactions
- This view of the Designing System provides links downstream to fully link PLM

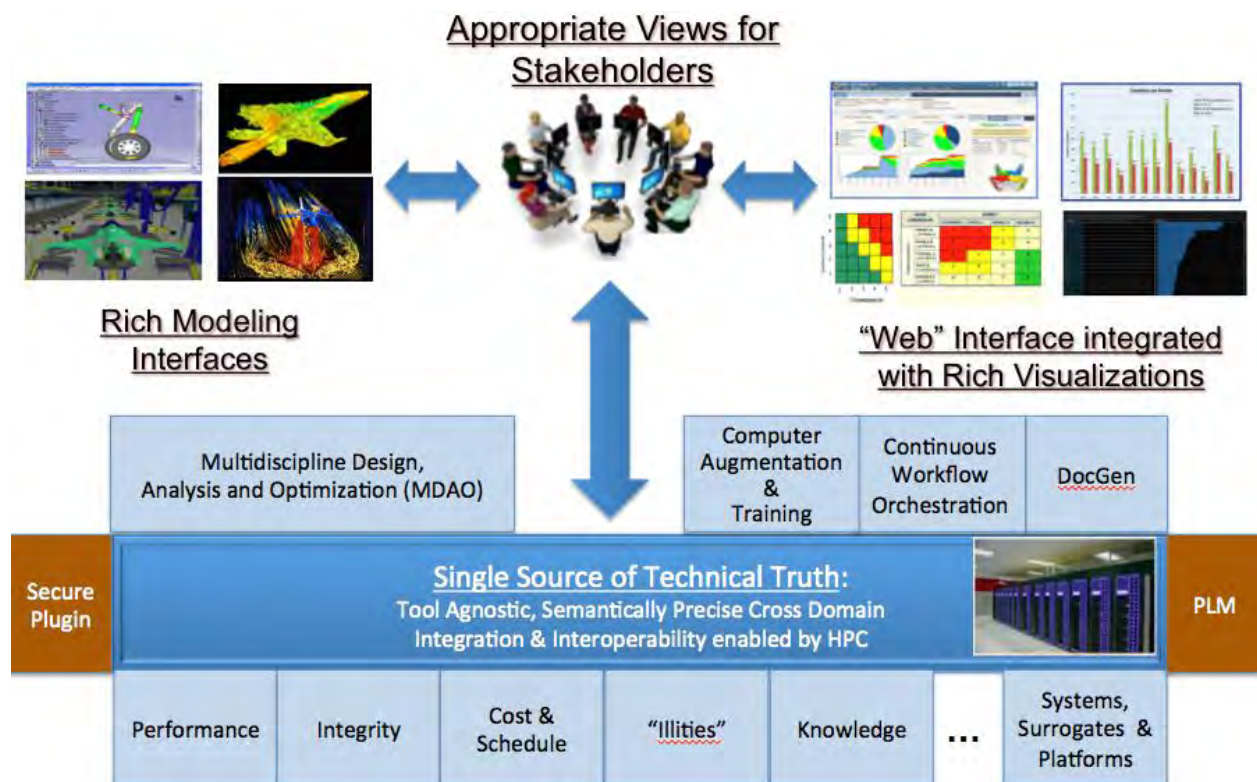


Figure 28. Integrated Environment for Iterative Tradespace Analysis of Problem and Design Space

## TASK 4 – INTEGRATED FRAMEWORK FOR RISK IDENTIFICATION AND MANAGEMENT

---

We are researching strategies, methods, and tools for a risk-based framework that aligns with the Vision model concept through MCE. This involves how the Vision model should include integrated risk identification and management. While there are many classes of risks to manage, for NAVAIR there are fundamentally two key classes of risk that we have been asked to consider:

- Airworthiness and Safety (most critical in Technical Feasibility assessment)
- Program execution (cost, schedule and performance)

There are also two complementary views of model-based acquisition with respect to risk:

- Risks introduced by modeling deficiencies and risks reduced by enhanced use of modeling
- Modeling to predict or assess risks (i.e., modeling for uncertainty quantification in acquisition and in the use of models)

The risk framework must also address the sponsor's question:

If we are going to rely more heavily on model-centric engineering, with an increasing use of modeling and simulations, how do we know that models/simulations used to assess “performance” have the needed “**integrity**” to ensure that the performance predictions are accurate (i.e., that we can trust the models)?

This brings in the need for approaches to what has been traditionally referred to as Verification, Validation and Accreditation (VV&A) of modeling and simulation capabilities. VV&A, in principle, is a process for reducing risk; in that sense VV&A provides a way for establishing whether a particular modeling and simulation and its input data are suitable and credible for a particular use [47]. The word tool qualification [48] and simulation qualification [106] have also been used by organizations regarding the trust in models and simulations capabilities.

There is also a concern that the risk of SE transformation to MCE will fail to provide an efficient, effective and reliable alternative to the current process. This is an important subject, but not address in this section.

This sections covers many elements needed for a risk framework, but is not exhaustive, such as:

- Risk context for the other topics covered in this section
- Risk consequences from model centric engineering
- Scope of the risk framework, which is fundamentally based on transitioning from document-centric processes to model centric engineering in assessing risk
- Model validation and simulation qualification
- Modeling, methods and tools for quantification of margins and uncertainty
- Risk-informed predictive models for risk identification based on subjective information
- Examples for improving the integrity of model
- Model-centric methods and tools enabling approaches to safety and airworthiness
- Risk in a radically transformed and collaborative environment
- Risk-related research

---

### RISK CONTEXT

Defined in the DoD Risk Management Guide [46],

Risk is a measure of future uncertainties in achieving program performance goals and objectives within defined cost, schedule and performance constraints. Risk can be associated with all aspects of

a program (e.g., threat, technology maturity, supplier capability, design maturation, performance against plan). Risk addresses the potential variation in the planned approach and its expected outcome.

Risks have three components:

- A future root cause (yet to happen), which, if eliminated or corrected, would prevent a potential consequence from occurring,
- A probability (or likelihood) assessed at the present time of that future root cause occurring, and
- The consequence (or effect) of that future occurrence.

A future root cause is the most basic reason for the presence of a risk. Accordingly, risks should be tied to future root causes and their effects. A risk framework needs to address how MCE can identify future risk and characterize its margins and uncertainty in the face of continual change of the problem analysis and design.

---

#### **RISK OF CONSEQUENCE FROM MODEL CENTRIC ENGINEERING TRANSFORMATION**

A concern is risk of adverse consequences resulting from radical transformation to MCE acquisition. Possible adverse consequences of concern are (a) failure to produce aircraft that can be certified as safe and airworthy, (b) failure to be able to certify airworthiness and safety, and (c) certifying unsafe or unworthy systems as safe and airworthy. We are not addressing the risk that MCE transformation fails to produce the desired reduction in acquisition time and cost.

We assume that radical transformation to MCE acquisition will **not** involve radical change to the airworthiness certification criteria (e.g., MIL-HDBK-516B/C [80]), or system safety goals, objectives and analysis framework. However, we do believe that the production of the evidence needed will be done in a very different way derived primarily from models and the associated analytical means.

We assume that transformation to MCE will have several major effects on the airworthiness and safety certification process. We assume that manual reviews and analyses of paper-based requirements, design, engineering and manufacturing documentation will be replaced with analysis of executable models and analysis using executable/dynamic models (i.e., analysis of the models, and analysis with the models) with interactive visualizations [41]. We assume that test design and analysis, at all levels of the system, will be conducted in an iterative process in which models will be used to define the conditions for the next test (experimental design) and to analyze the test results. We assume that models of the system, models of the test process and instrumentation, and models of the uncertainty in the system models will be used to define tests that will produce the greatest possible reduction in (a) uncertainty regarding airworthiness and safety, and (b) reduction in uncertainty with regard to the validity of the models and the inputs to the models. We assume that results of the testing will be used to refine the models and their calibration data, as well as being used to score the system with respect to airworthiness and safety certification criteria.

The risk assessment framework consists of identifying the major areas or types of risks resulting from transformation to MCE, and assessing whether those risks are manageable (i.e., the feasibility of effective risk management). Risk management consists of identifying risks, quantifying, planning and implementing, detecting, mitigating, and monitoring detection and mitigation, and estimating uncertainties in them.

Some major risk types and areas that we identified are:

- Models do not have adequate resolution, completeness and fidelity to be used to address the airworthiness and safety criteria
- Models do not have adequate fidelity with respect to the manufacturing process, manufactured test articles and test implementation
- Models of human behavior and performance are inadequate with respect the range of human errors and processing limitations, and simulators of man-in-the-loop testing fail to adequately simulate the phenomena in the operating environment
- Adaptive nature of the iterative model-test-model cycle leads to homing in on specific areas of uncertainty while avoiding/ignoring others
- Unstated assumptions in the airworthiness criteria and in the models are inconsistent and incompatible
- Process of model calibration and validation with respect to airworthiness and safety concerns requires the same procedures, tests, reviews and analyses as the current airworthiness and safety process to achieve the same level of certainty
- Model-centric airworthiness and safety certification will require more effort and a different skill set than the current process
- Model-centric approach will conceal “blind spots” – factors and effects not included in the models will be ignored or concealed in certification, test design and analysis
- Calibration and validation strategies for highly non-linear events and limited test & observation opportunities
- Models used out of context, outside validation & calibration
- Limitations, assumptions, and phenomena omitted, not often not well articulated
- Deterministic chaos phenomena where small change in boundary conditions (inputs) produces rapid divergence in outputs not reflected in models or simulation scenarios (non-linearities)
- Sensitivity to complex and often unknown boundary conditions
- Gaps in understanding multi-scale, multi-physics phenomena, potentially due to limitations in cross-domain model integration
- Human behavior, knowledge, cognition – flight safety, damage control
- Level of modeling and simulation different from level of analysis and decision
- Incompatible scope, resolution, terminology with test procedures

The standard for acceptance is that the model-centric process is not worse, not less reliable, than the current process in any area or aspect of airworthiness and safety certification.

It is our opinion that these identified risks are potentially manageable. However model calibration [78], validation and accreditation for MCE with respect to airworthiness and safety may require significant effort and expertise [47]. The airworthiness certification handbook (and its expanded version), and lessons learned from previous airworthiness and safety assessments provide detailed, but incomplete, insight into the resolution and fidelity needed in the models. There has been significant progress in high-resolution man-in-the-loop simulators, airworthiness compliance verification via simulation, and formal model verification and completeness processes.

---

#### **FUTURE ROOT CAUSES**

As the focus of the effort is on understanding the problem, including pre-milestone A through CDR, it will be important to understand MCE approaches to assessing the potential future root causes of risk especially as the adversaries are attempting to leverage unexpected future concerns, for example:

- Adversaries adapt to avoid our systems' strengths and exploit their limitations by their choice of battlefields, tactics, and equipment
- "Long-Lived" DoD Systems
- Systems design to be adapted to counter adversary adaptations and exploit maturation of our emerging technologies
- To deter and defeat current threats
- To enable cost-effective upgrade & adaptation

This is not an exhaustive list.

---

## SCOPE OF THE RISK FRAMEWORK

We worked with our NAVAIR team members to determine the scope for the risk framework. Key to the representation of the models (and Task 3) to support risk identification and management is to characterize the **types of evidence** that are required for **Flight clearance and Flight readiness**. It is important to understand how the models are developed and derived in order to understand the risk strategies that must be in place for **identifying and assessing the evidence for flight clearance**.

The process for risk under consideration for this SE transformation covers system development from Milestone A to CDR (at least for now). These questions related to risk also helped to refine the scope for Task 3, and introduced a new term Digital CDR (DCDR), with a heavy emphasis on **digitally-derived evidence for airworthiness and safety, but to also include program execution**.

In both preliminary discussions with organizations and our NAVAIR team, it is recognized that it is important to quantify "margins" and "sensitivities" and "uncertainties" as a way to quantify risk.

As an example, one of the organizations (in our preliminary Task 1 discussion) creates new types of advanced material for a system. They cited a particular effort working with advances in new material and processes at the nanoscale. At the component level the margins seemed acceptable. However after composing the components, margins propagated to unacceptable levels in the final integrated form.

Risk implies probabilities of what might go wrong or might not happen (on time or due to the degree expected), and some distribution of magnitude of consequences. This requires "impossible certainty" of the degree of uncertainty and advance knowledge of the likelihood and effects of unidentified events and factors. Therefore, we suggested that a better framework might be to work in terms of design margin. Design margin is more closely related to design. Design margin is how much room there is for a subsystem or component to perform less well than expected or to have greater burdens than expected until it becomes a problem. In some cases, e.g. weight, any increase adds to total weight, so instead of a weight margin, we might want to think in terms of sensitivities (sensitivity in increase in total weight, time, cost, etc. to a percentage increase in the component weight, time, power draw, etc.). This creates a number of questions for this task, for example:

- Can we use models to see how much design margin there is in a system – specifically when we cannot push the system to failure; which types of models and how can we use them to estimate the conditions under which the system begins to exhibit unstable response
  - In control systems analysis this is often taken to be the 3dB point – the frequency of input variation at which the output-to-input ratio is half what it was for low frequency change, or the 90-degree phase-shift point, where the frequency of input variation at which the system response lags by 90 degrees

- Control systems analysis methods also address the acceleration, velocity and displacement limits at which the system dynamics change
- Failures are often associated with transitions from linear to highly non-linear regimes; often the structure, interactions and/or dynamics change in these regions (e.g., insulators or isolators fail, etc.) – e.g., the acceleration, velocity and displacement limits at which the system transitions from linear to non-linear response
- Models that are relevant in the “linear” regime will give erroneous results in the non-linear regime
- Models that do not represent the dynamics that change the structure of a system (e.g., insulation wearing off causing a short-circuit, structural failure of a linkage, strain transitions from elastic to plastic deformation, etc.) will give erroneous results

Mechanical or electro-mechanical control and isolation systems are good examples, and important for airworthiness. Control systems work within a limited range. Standard control system analysis examines the frequency response and looks for the 3dB frequency, i.e. the frequency at which the transfer function is half of the low-frequency value (the transfer function is just the ratio of output-to-input). Other limits include maximum displacement, velocity and acceleration – when the system hits hard-stops, current limits etc.

Surrogates can be driven with increasing frequency inputs to find the 3dB point without having to experience the failure. The input parameters of virtual models are often “tuned” to match the 3dB point of test data, and then used to extrapolate to find the 3dB point of hypothetical systems. Physically realistic models can be used to estimate the limiting thresholds of stable response, provided the models and inputs are adequately calibrated and validated. Special consideration is needed for basic physical processes with non-linear response in the regime of operation, e.g., friction between moving parts versus friction between stationary parts.

Nested control loop models have been used effectively in system safety modeling and analysis [75]. The outer control loops detect changes in the response behavior of inner control loops, and then adjust the parameters of the inner control loops to bring the inner loops back into the stable regime.

In the use of modeling and simulation, there are different types of simulation with different levels of fidelity. A significant challenge is that tools do not often map well to different levels of abstractions. These are areas to frame risk. There are increasing uses of model transformation from one level or to different disciplines. Model transformation and model consistency between these views becomes a risk issue.

A companion concept is credibility of the estimates of performance, cost, etc. High credibility if it has worked in a surrogate system, less if it is similar to something demonstrated in a surrogate and model extrapolation. It will be important to better understand model extrapolations.

- Less credibility the farther the model extrapolation is extended
- Less credibility going from surrogate system to bench testing, etc.
- Use of multi-scale calibration and validation
- Use of progressive model-based design confirmation in technical reviews
  - Subsystems mature and are integrated at different rates
  - Sometimes early decisions are needed for long-lead time items whose specifications can be confirmed before other aspects of the system (e.g., final control system parameter values)



---

## MODELING AND METHODS FOR UNCERTAINTY QUANTIFICATION

Sandia National Laboratory discussed some advanced approaches for supporting uncertainty quantification (UQ) to enable risk-informed decision-making. Their methods and tooling address the subjects of margins, sensitivities, and uncertainties. The information they provided reflects on the advanced nature of their efforts and continuous evolution through modeling and simulations capabilities that operate on some of the most powerful high performance computing (HPC) resources in the world. We heard about their HPC capabilities, methodologies on Quantification of Margins and Uncertainty (QMU), an enabling framework called Dakota, and the need and challenge of Model Validation and Simulation Qualification [106]. They also discussed the movement towards Common Engineering Environment that makes these capabilities pervasively available to their entire engineering team (i.e., the designing system in our terminology). We think their capabilities provide substantial evidence for the types of capabilities that should be part of the risk framework. This section provides additional details.

---

### DAKOTA SENSITIVITY ANALYSIS AND UNCERTAINTY QUANTIFICATION (UQ)

The Dakota framework supports optimization and uncertainty analysis [109]. There is significant demand at Sandia for risk-informed decision-making using credible modeling and simulation:

- Predictive simulations: verified, validated for application domain of interest
- Quantified margins and uncertainties: random variability effect is understood, best estimate with uncertainty prediction for decision-making
- Especially important to respond to **shift from test-based** to modeling and simulation-based design and certification
  - This gets to an important point about how to use models as opposed to testing, which is critical for NAVAIR's objective to rapidly and continuously "cross the virtual V"

The HPC capabilities comes into play as they are built to take advantage of the HPC environment and can be combined with predictive computational models, enabled by environment and culture that focuses on theory and experimentation to help:

- Predict, analyze scenarios, including in **untestable regimes**
- Assess risk and suitability
- Design through virtual prototyping
- Generate or test theories
- Guide physical experiments

Dakota is referred to as a framework, because it is a collection of algorithms supporting various types of integration through programmatic (scripting) interfaces; this is representative of the concept of model-centric engineering, see Figure 29. It automates typical "parameter variation" studies to support various advanced methods and a generic interface to simulations/code, enabling QMU and design with simulations in a manner analogous to experiment-based physical design/test cycles to:

- Enhances understanding of risk by quantifying margins and uncertainties
- Improves products through simulation-based design
- Assesses simulation credibility through verification and validation
- Answer questions:
  - Which are crucial factors/parameters, how do they affect key metrics? (sensitivity)
  - How safe, reliable, robust, or variable is my system?  
(quantification of margins and uncertainty: QMU, UQ)

- What is the best performing design or control? (optimization)
- What models and parameters best match experimental data? (calibration)

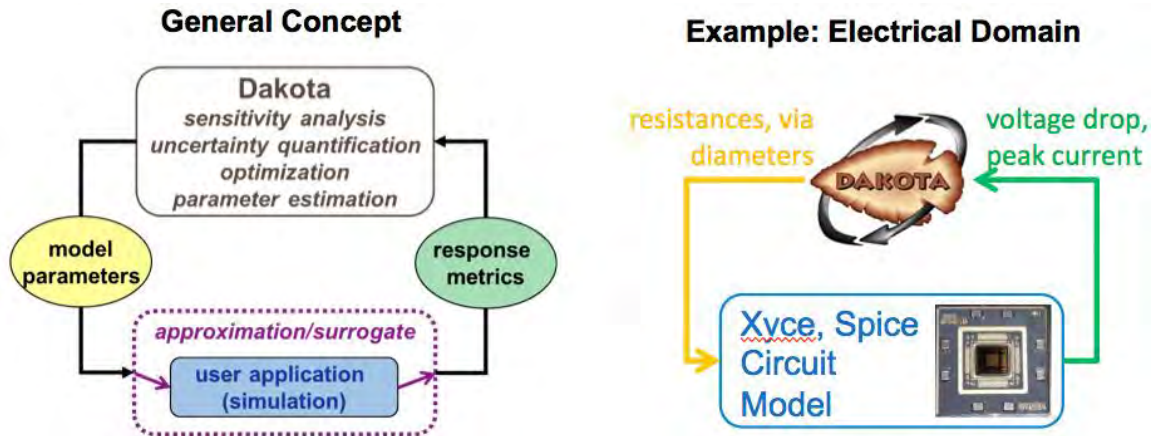


Figure 29. Dakota Framework Integration Wraps User Application

To put margins and uncertainty into context, assume that there is a device that is subject to heat, and we need assess some type of thermal uncertainty quantification. Given some results from some Design of Experiment (DoE) (also supported by Dakota) results that give a probability distribution as shown in Figure 30 [2]. The Mean of the temperature:  $T$ , to the lower bound of the threshold (e.g., 72 degrees) characterizes the Margin, and the Standard Deviation ( $T$ ) characterizes the uncertainty.

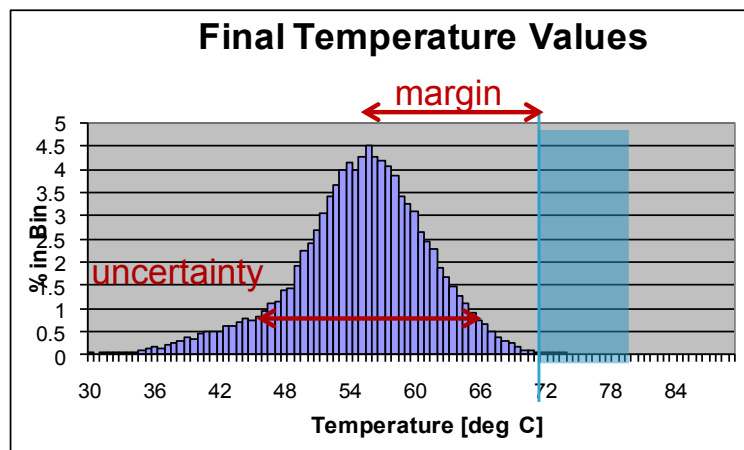


Figure 30. Example for Understanding Margins and Uncertainty

This approach and Dakota framework supports a broad set of domains, and therefore we think it can be generally applied across domain for NAVAIR, for example:

- Supports simulation areas such as: mechanics, structures, shock, fluids, electrical, radiation, bio, chemistry, climate, infrastructure
- Is best used with a goal-oriented strategy:
  - Find best performing design, scenario, or model agreement
  - Identify system designs with maximal performance
  - Determine operational settings to achieve goals
  - Minimize cost over system designs/operational settings
  - Identify best/worst case scenarios

- Calibration: determine parameter values that maximize agreement between simulation and experiment
- Handles parallelism, which is often not feasible with commercial tools, and why HPC can play an important role
- Provides sensitivity analysis – find the most influential variables
- **Uncertainty Quantification**
  - Models inherently have uncertainty
  - Assess effect of input parameter uncertainty on model outputs
    - Determine mean or median performance of a system
    - Assess variability in model response
    - Find probability of reaching failure/success criteria (reliability)
    - Assess range/intervals of possible outcomes

---

#### AN OVERVIEW OF QUANTIFICATION OF MARGINS AND UNCERTAINTY

Dakota is a tool framework that can support the method of Quantification of Margins and Uncertainty (QMU). Some of the material from Sandia is categorized “Official Use Only [OUO].” We provide a summary extracted from publically available information [87].

QMU pre-dates Dakota and is not unique to Sandia as it was used at Lawrence Livermore National Laboratory and Los Alamos National Laboratory, with the original focus of the methodology to support nuclear stockpile decision-making<sup>22</sup>. QMU is a physics package certification methodology and although it has been around and used at Sandia dating back to 2003, and both QMU theory and implementation are still being developed/evolved [87]. We believe the methodology has more general use than just physics package certification.

QMU applies to the lifecycle of the entire weapon, with focus on:

- Specification of performance characteristics and their thresholds
  - Performance is the ability of system/component to provide the proper function (e.g., timing, output, response to different environments) when exposed to the sequence of design environments and inputs
- Identification and quantification of performance **margins**
  - A performance margin is the difference between the required performance of a system and the demonstrated performance of a system, with a positive margin indicating that the expected performance exceeds the required performance
- Quantification of uncertainty in the performance thresholds and the performance margins as well as in the larger framework of the decisions being contemplated

There are two types of uncertainty that are generally discussed that account for, quantify, and aggregate within QMU:

- Aleatory uncertainty (variability)
  - Variability in manufacturing processes, material composition, test conditions, and environmental factors, which lead to variability in component or system performance
- Epistemic uncertainty (lack of knowledge)
  - Models form uncertainty, both known and unknown unknowns in scenarios, and limited or poor-quality physical test data

---

<sup>22</sup> The Comprehensive Nuclear Test Ban Treaty ends full-scale nuclear weapons testing in the U.S. President Bill Clinton at the United Nations, September 24, 1996

The statistical tolerance interval methodology is an approach to quantification of margins and uncertainties for physical simulation data. There is also probability of frequency approach, commonly used in computational simulation QMU applications [87], which:

- Extends the “k-factor” QMU methodology for physical simulation data
  - k-factor, in general, is defined as margin divided by uncertainty (M/U)
    - Margin (M): difference between the best estimate and the **threshold** for a given metric
    - Uncertainty (U): the range of potential values around a best estimate of a particular metric or threshold
  - Provides essential engineering analysis to ensure the collected data sample includes measurements that may be used to infer performance in actual use
  - It is important to understand the performance requirement to understand the **performance threshold and associated uncertainty**
    - Threshold: a minimum or maximum allowable value of a given metric set by the responsible Laboratory
- The new method addresses the situation where performance characteristic has shown the potential for low margin or a margin is changing (likely getting smaller or there is greater uncertainty) with age [87]
  - Notionally the margin shifts from the mean of the performance characteristic (PC) and its performance requirement (PR) to the difference between a meaningful percentile of the distribution of the performance characteristic and its performance requirement
  - Need to quantify uncertainty through the computation of a statistical confidence bound on the best estimate of the chosen percentile rather than by a sample standard deviation (as reflected in Figure 30), which does not account for sampling variability
  - This is accomplished by computing a statistical tolerance interval

We created a graphic from several publically available sources, as shown Figure 31 in order to better explain a few aspects about QMU, Dakota, epistemic and aleatory uncertainty. Typically within the Dakota framework there is an outer loop: epistemic (interval) variables and inner loop: uncertainty quantification over aleatory (probability) variables (e.g., the probability distribution). The outer loop determines interval on statistics, (e.g., mean, variance). The inner loop uses sampling to determine the responses with respect to the aleatory variables. This information can be used to understand the epistemic and aleatory uncertainties, relative to the Lower Performance Requirement (LPR).

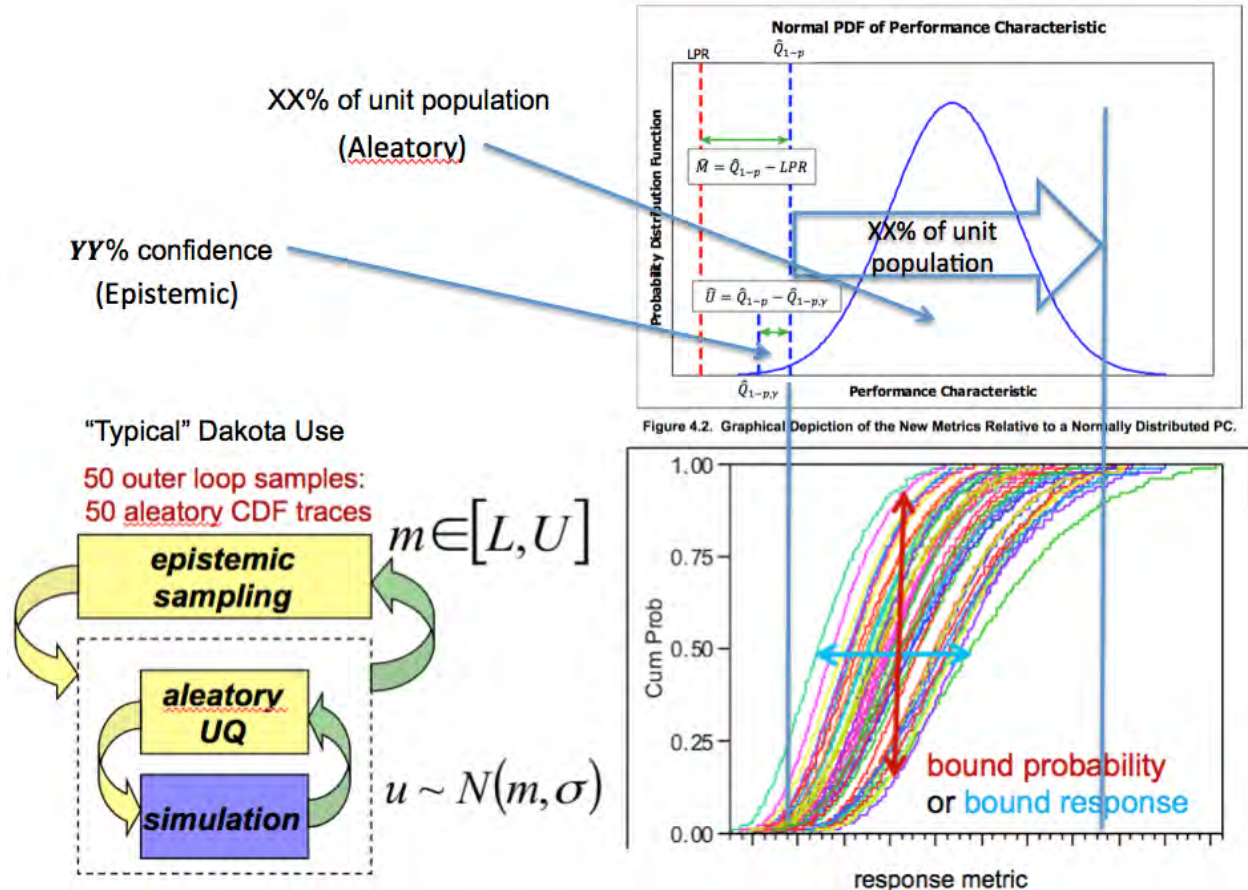


Figure 31. Pulling Together Concept Associated with QMU

The information is relevant to the risk framework as it provides evidence about methodologies and tools to deal with several of the topics. QMU and Dakota are still evolving, and there are a number of challenges:

- How do we ensure that we use the right “data” as inputs?
- How to roll up to the system level?
- Model validation and simulation qualification

#### RISK FRAMEWORK APPROACH TO UNCERTAINTY MODELING AND PREDICTION

The SERC team has also been working with NAVSEA to develop a framework and approach to uncertainty quantification modeling and prediction. The approach has three main components:

- Identifying the design, test and modeling factors at different system scales
- Analyzing the uncertainty, variability, and error in design implementation, testing, and modeling
- Using experimental design methods to assess the contributions and interactions to system (airworthiness and safety) and program execution risks

The risk modeling and analysis approach also addresses potential errors and uncertainties in the overuse of limited data. Ideally:

- One data set is used to identify critical factors
- A second independent data set is used to develop the models

- A third independent data set is used to calibrate the models
- A fourth independent data set is used to assess the expected error in model results

In practice data sets, surrogate vehicle test data, etc. are limited. Bootstrap methods use repeated resampling of the data and repeating the modeling and analysis process to obtain a statistical estimate of the uncertainty in model-based acquisition given the available data. Further analysis reveals the value – reduction in uncertainty – for additional data.

These types of models capture and embed knowledge associated with expert judgment, historical evidence and rules of thumbs that are used in the decision-making process. Alternative methods help deal with these type of issues.

---

## **PREDICTIVE MODELS FOR RISK**

There are situations where we do not have good historical quantitative data and we often use expert judgment. This section discusses a predictive modeling approach when risk involves subjective information, small data sets, and “dirty” data.

The SERC team has developed and used models in the prediction of risk, and plans to use predictive analytic models to support risk identification and management. More generally we can use models to provide risk quantification for almost all types of decisions that are made by stakeholders (e.g., model-based reviews). As an example, we created a Bayesian model using factors derived from the Airworthiness standard MIL-HDBK-516B [45] as shown in Figure 32. This is conceptually similar to the approach we are using on an FAA NextGen research task for collaborative risk-informed decision-making [15][16][17]. The key characteristics of the approach are they ensure that all factors are considered in the decision-making process, and that all classes of stakeholders are adequately represented in the decision-making process. A systematic and comprehensive treatment of all relevant factors provides better risk identification.

We used this model and an example from a true story related to a C130 Weapon Delivery system to illustrate the concept. While this model is notional at this time, this example started a discussion with the team about how stochastic (probabilistic) models can play an important part of the Vision as they formalize many aspects of the human decision making process that will be important at many gates, reviews, and decision points of the Vision concept. Each factor covers a specific aspect of airworthiness, to ensure that all possible uncertainties and risk are considered in the quantification of risk. The risk index is a probability distribution, where for example, the mean can map to quantities in a risk matrix.



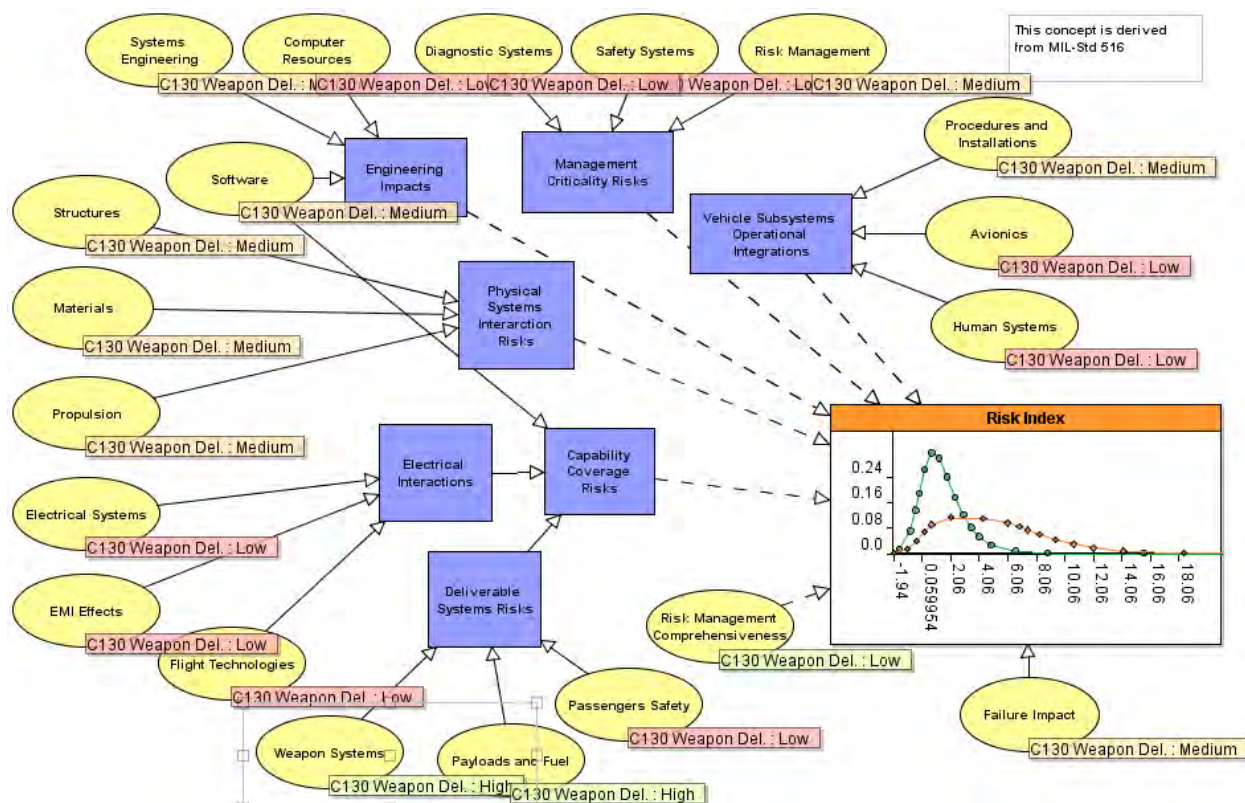


Figure 32. Bayesian Model Derived from Airworthiness Factors

#### RISK FRAMEWORK CAPTURES KNOWLEDGE

These types of risk frameworks are actually knowledge models of credibility (not models of performance, but models of uncertainty). Part of the effort on modeling the “As Is” process (Task 3) is to identify and then formalize within the models the information and associated knowledge for evidence-based decisions and evidence-based timing of decisions. Other considerations and opportunities:

- In the “As Is” process, what decisions are artifacts of the process, but not essential to the engineering development?
- Are there lost opportunities by making early concept and design decisions?
- Is there a risk of bad decisions, risks and costs of no or deferred decisions, during planning, or during execution?
- Reconsider the “full system” technical review model. Not all parts of the system are ready for PDR, CDR at the same time. Some are more mature than others. Maybe a granular approach is needed.

The timing of technical reviews and decisions should be made when there is an accumulation of evidence sufficient to make a credible decision. Ideally, this will be inherent in the Vision concept, when the required information and associated analyses are complete, the evidence and timing for decisions should be triggered events in the automated workflow.

---

## MODEL VALIDATION AND SIMULATION QUALIFICATION

Comparing model predictions to observed responses for the purpose of assessing the suitability of a particular model constitutes what is known as *model validation*. Uncertainty quantification for simulation models is not strictly limited to model validation. When experimental observations are available for validation assessment, analysts would often like to use the same observations for *model calibration*, which is the process of adjusting internal model parameters in order to improve the agreement between the model predictions and observations. However, if internal model parameters can be adjusted in this manner, this means that there is some amount of uncertainty associated with the true, or best, values of these parameters. Uncertainty associated with model inputs directly implies uncertainty associated with model outputs [77].

Model validation and simulation qualification are ways to ensure that “integrity” of the models prediction information. Sandia has developed the “Real Space” model validation approach [103], which was formulated by working backwards from an end objective of “best estimate with uncertainty” (BEWU) modeling and prediction, where *model validation* is defined as: the process of determining the degree to which a computer model is an accurate representation of the real world from the perspective of an intended use of the model. However, the interpretational and implementation details can still vary widely.

We have discussed a number of model validation and simulation qualification topics, such as:

- Hierarchical Model Validation
  - Seeks to expose key physics and material models that are brought together, and asks are the combined products validated at various levels of aggregation? “right for the right reasons”
  - Seeks to catch interactions and emergent behaviors not present in validation of separate models
  - Also need to consider “Traveling” or “Linking” variables that bridge modeling levels [106]
- “Exercising” the models at the “boundaries” of the probability distributions (~10 and 90 percentile)
  - This is related to a recommended testing strategy based on boundary-value analysis/testing (i.e., exercising the “element under test” at the boundaries can expose more anomalies than exercising the nominal/typical tests scenarios)
  - Has greater potential to expose off-nominal cases

Various model validation paradigms and methodologies are still being proposed, developed, and tested. There is no overriding consensus exists yet on “best” approach. We questioned Sandia about an idea that we had in our working session about how we are increasing in the ability to do more “integration” of the simulation across domains, and can that “integration” provide increased visibility into potential anomalies, therefore allowing us to better understand the “integrity” of the simulations. This is analogous to why integration testing often exposes issues

Sandia provided some papers that we can share with the team [106]. This information provides significant guidance and historical perspectives that should be further used to support the concept of model validation and model integrity as part the Vision for Task 3. In addition, Sandia discussed the Predictive Capability Maturity Model (PCMM), which is an evolving model that can be used to assess the level of maturity of computational modeling and simulation (M&S) efforts [88]. The PCMM addresses six contributing elements to M&S: (1) representation and geometric fidelity, (2) physics and material model

fidelity, (3) code verification, (4) solution verification, (5) model validation, and (6) uncertainty quantification and sensitivity analysis.

There is more research planned for a follow-on phase of this research. Here are other topics that have discussed related to improving our trust in models and simulation:

- Numerical integration techniques [57]
  - This is an example provide by NASA/JSC related to simulation of space vehicles for different planetary bodies
  - Propagating the evolution of a vehicle's translational and/or rotational state over the course of a simulation is an essential part of every space-based Trick simulation. The underlying equations of motion for this state propagation yield second order initial value problems. While analytic solutions do exist for a limited set of such problems, the complex and unpredictable nature of the forces and torques acting on a space vehicle precludes the use of analytic methods for a generic solution to these state propagation problems. Numerical integration techniques must be used to solve the problem.
- Flights validate models/simulations
- Use logged data to continually calibrate models/simulation
  - This was discussed in our organizational visits, and it was discussed as part of model guidance
  - Model calibration should be getting easier, because we have better data collection, storage, and the ability to analyze large data sets
- Models of pedigree
- Model Validation Review (MVR)
- Cross-domain integration of models may also be a way to have greater confidence in simulation models
  - We know that integration and integration testing often exposes many defects or anomalies
  - We currently do not have much cross-domain integration of models/simulation
  - These are new capabilities and the inherent nature of model-centricity will lead to greater integration; this could potentially provide new types of inputs/measures (insights) to help us build trust in the models
- Probabilistic Risk Analysis – this might be yet another related cross-domain approach
  - Organization discussed an example related to using simulation and Dakota to reduce the number of flight tests
- Bayesian model calibration [76]
  - Model calibration is a particular type of inverse problem in which one is interested in finding values for a set of computer model inputs, which result in outputs that agree well with observed data
- Bayesian-based qualification planning [103]

Finally, Bill Brickner from NAVAIR points out that no mission-level model can ever be validated – that is, it is being used to predict possible future scenarios. We will continue to investigate approaches.

---

## IMPROVING THE INTEGRITY OF MODELS

The DARPA META program discussed the issues and therefore risks with attempting to use models that were not developed according to a methodology that results in models that are suitable to the capabilities of the tools [7]. Increasingly, there are approaches being used to improve the integrity of the models through various types of checks, for examples:

- NASA/JPL use an ontology-based approach for ensuring models are developed to comply with modeling patterns, and have developed over 60,000 checks using computationally based formal reason to ensure the models are well-formed, consistent and complete with respect to the modeling pattern [69]
- Formalized evaluation criteria of a modeled system architecture by encoding 23 axioms in two different tools: Vitech CORE and Innoslate from SPEC Innovations to demonstrate concept [104]
  - Some tools provide pre-defined checks, but in this case the tools allow for additional types of checks to be performed that can be adapted to a particular methodology
- Apply an approach for Verification, Validation, and Accreditation Shortfalls for Modeling and Simulation based on criteria associated with modeling and simulation patterns in SysML [96]
- DARPA META also developed a concept of Probabilistic Certificate of Correctness (PCC) [7]
  - PCC is used to verify that the system will perform within the requirements bounds within a probabilistic certainty factor
    - PCC uses statistical sampling techniques on system and environmental parameters, executed on test benches for evaluation of the statistical properties of the metrics
    - Requirements objectives and thresholds define the numerical bounds of the acceptable metric values, and a probability of the design metric existing within the bounds is computed
  - Variation in performance is expected due to manufacturing variations of components and possibly environmental conditions

A risk framework needs to consider these and other types of checks. These types of checks not only ensure that the model is well-formed, consistent and complete with respect to modeling patterns, but this also reduces risk, because these types of rules provide greater assurance from the use of tools that use these models for simulation, analysis and generation.

---

## **MODEL-CENTRIC METHODS AND TOOLS ENABLE APPROACHES TO SAFETY AND AIRWORTHINESS**

The emerging approach to Failure Modes and Effects Analysis (FMEA) and safety analysis is to use formal, automated methods to derive the FMEA and safety analysis models from the SE model of the system, instead of the historical practice of developing informal models by manual means, and to be able to integrate failure and safety analysis earlier in design. The reliance on the SE model of the system potentially exposed the failure mode and safety analysis to the risk of an incomplete or incorrect systems engineering model. This risk can be mitigated by:

1. Using formal methods based on a meta-model in developing the system model
2. Applying automatic model verification methods to the system model
3. Applying error and uncertainty propagation methods to analyze the accuracy of the system model

These are areas of active research and development, and the methods, tools and techniques are rapidly maturing. The current state-of-the-art is that there are available methods with demonstrated capability, but not complete capability or entirely turnkey operation. Research and development is focusing on integrating software FMEA with hardware FMEA, and higher-level resolution of failure modes.

We believe that formalization of MCE can lead to advances aligned with the following hypotheses:

- H1. Model-based methods to automatically generate Failure Modes and Effects Analysis (FMEA) models tools (1) are feasible and practical, (2) are superior to traditional manual methods, (3) can address software failure modes, and software-hardware interactions in cyber-physical systems.

- H2: Model-based safety analysis methods, procedures and tools (1) are feasible and practical, and (2) are suitable for analysis of mission critical requirements and design in the phases of the acquisition lifecycle prior to flight certification testing.
- H3: Risks in reliance on models are manageable, and model analysis methods provide a level of assurance of model correctness and accuracy.

We provide a summary of some research finding in support of the hypotheses.

---

### **MCE METHODS TO GENERATE FAILURE MODES AND EFFECTS ANALYSIS (FMEA) MODELS**

Model-based methods to automatically or semi-automatically generate Failure Modes and Effects Analysis (FMEA) models from Systems Engineering (SE) models of requirements, function, design and their interaction are feasible, practical, and have been demonstrated. There are alternative technical approaches to enhancing the system model to automatically identify abnormal modes and chains of effects. The automated approaches do not address the criticality of different system functions, only how the functions are degraded. The automated modeling approach provides higher resolution of degraded mode states for high-level functions as a combinatorial analysis of low-level failure states. The SE models need to conform to a meta-model framework for automating FMEA models.

Schindel describes a systematic approach beginning with a “metamodel” to organize and express the system requirements and high-level design MBSE model [110]. The MBSE model defines the system architecture, functional requirements and functional dependencies. The system architecture is the organization of the system into subsystems down to a “black box” level, and the interconnections or links between elements. Functional requirements interaction behaviors and performance under specified conditions in an objective technical language for the system and every subsystem down to the “black-box” resolution of the MBSE model, and for each of the links between elements. Conditions are external and internal states of the system and its internal subsystems. Behaviors are state changes. Functional dependencies are the hierarchical relationships among functions, in which higher-level functions depend on lower level functions. Requirements are automatically converted into counter-statements that define the failure modes – failures of system elements and links to perform one or more of their functions. The effects are automatically traced through the links and nodes in the system architecture, and propagated up through the functional hierarchy.

Increasing complexity makes manual methods laborious, and highly depends on the experience and expertise of the practitioners. Manual approaches are subject to human error of omission, and are not tightly linked to the system model. These issues are addressed with automated FMEA model generation from the Systems Engineering model. Automated FMEA model generation creates the possibility to extend FMEA to a comprehensive analysis of degraded mode states. The automated methods do not distinguish between important (likely and significant) failure modes in developing the FMEA model – a process guided by the insight and experience of the practitioners in manual FMEA modeling. The FMEA models generated by automatic methods are too large to understand by reviewing the model. Automated analysis methods are needed. These methods are still under development. Automated analysis requires input data regarding the likelihood or frequency of the different failure modes, and the importance of high-level function degradation mode. Manual models require the same data, but since the data are often not available (just subjective estimates), the dependencies are built into the model based on “expertise and experience.”

Automated FMEA analysis can address software failure modes, and software-hardware interactions in cyber-physical systems. FMEA addresses the behavior and response of the system to changes in (a) external conditions (environment, demand), and internal conditions (performance time, accuracy and



error of functional processes and the state or mode of logical subsystems). State transition modeling is a “best practice” to specify the system behavior and to test the behavior specifications. Automated FMEA for software requires a model of the logical states and transitions of the information in the system, as well as the physical states and transitions of the system and its environment. In real-time cyber-physical systems, timing of effects of sensing, processing, signaling and response on transitions needs to be included.

There are many possible failure causes and modes. The nominal behavior as specified in the state transition model may not be the desired behavior, i.e. the conceptual design may not satisfy the requirements, or can exhibit unplanned behaviors (“sneak circuits” and unplanned feedback loops). The behavior specifications (the state transition model) may be incomplete, i.e., there are possible states and transitions that are not represented in the model, e.g., the effects of failure of an actuator (stuck or random response), noise or failure of a sensor, short-circuits, and unexpected combinations of logical information states. The system can have calibration errors, or not account for effects of temperature, pressure, etc. on the calibration. Discretization of time and signal level can lead to control systems exhibiting behavior different than expected from a continuous model. Software specifications can be implemented incorrectly, i.e., coding errors.

Snooke presents an approach for automated, model-based software FMEA [114]. He addresses three classes of software failures: abnormal input values to a routine, failure of the processing hardware, and logical/algorithmic/semantic error in the software code. He does not address incomplete or incorrect specifications for the software. His approach combines methods from software diagnosis (functional dependency modeling) with automatic fault tree methods.

Another approach for automated, model-based software FMEA demonstrates the method in an application to aircraft flight control software [126]. The approach is concerned with failure modes resulting from the state of logical information in the system and the transformation of information state by processing, including timing constraints and synchronization. The approach relies on a system requirements model of behaviors and performance under conditions, and a software requirements model of information states, state transformations, timing interactions, etc. They define a structured approach to system and software modeling, and for the use of the models to analyze software failure modes and effects. They applied the method to flight control software that controls the cabin door, the brake system and the front flap.

---

#### **MISSION-LEVEL MODEL-BASED SAFETY ANALYSIS**

Model-based safety analysis methods, procedures and tools are feasible and practical for mission critical safety assessment. System safety analysis methods are commonly based on informal system models. The lack of precise models of the system architecture and its failure modes often forces the safety analysts to devote much of their effort to finding undocumented details of the system behavior and embedding this information in the safety artifacts such as the fault trees.

NASA/JPL developed a comprehensive Safety-Driven, Model-Based System Engineering Methodology that enables system engineers to design systems from a safety point-of-view, i.e., with hazard analysis folded into the nominal design process rather than conducted as a separate activity [60]. This methodology integrates MIT’s Systems-Theoretic Accident Model and Processes (STAMP), STAMP-Based Hazard Analysis (STPA), intent specifications (a structured, constraint-based system engineering specification framework), and JPL’s State Analysis (a model-based systems engineering approach). The methodology was developed to address the challenges of safety design and analysis for complex, software-intensive cyber-physical systems, and to do so early in the design process.



Model-based safety analysis has been approached as a special case of automated, model-based FMEA approaches, except that instead of addressing all system requirements and their derived requirements, it is restricted to safety-related requirements and their derived requirements. Benefits of model-based safety analysis include:

- Tight integration between systems and safety analysis based on shared models of system architecture and failure modes
- The ability to simulate the behavior of system architectures early in the development process to explore potential hazards
- The ability to exhaustively explore all possible behaviors of a system architecture with respect to some safety property of interest using automated analysis tools
- The ability to automatically generate many of the artifacts that are manually created during a traditional safety analysis such as fault trees and FMEA/FMECA charts

Joshi and Heimdahl leverage existing tools and techniques from model-based development to create formal safety models using tools that are familiar to engineers and the static analysis infrastructure available for these tools [70]. They enhance the system model with a fault and effects model to automate much of safety analysis, and do so in a way that the safety issues point back to design elements that produced the adverse behavior. Their approach composes a model of nominal system behavior to model of failure modes and effects on behavior under a set of common malfunctions and failure modes of elements of cyber-physical systems. Their approach uses automated model checking methods that will either verify safe behavior in the presence of “N” faults, or find a combination of “N” faults that produce adverse behavior (i.e., failure to meet a safety critical functional requirement). They apply the method to the aircraft wheel braking system example from ARP4761 [108].

HiP-HOPS (Hierarchically Performed Hazard Origin and Propagation Studies) is an automated model-based method for safety analysis that enables integrated assessment of a complex system from the functional level through to the low level of component failure modes [94]. The failure behavior of components in the model is analyzed using a modification of classical FMEA called Interface Focused-FMEA (IF-FMEA). One of the strong points of this approach is that the fault tree synthesis algorithm neatly captures the hierarchical structure of the system in the fault tree.

---

#### **RISK RELIANCE ON MODEL CORRECTNESS AND ACCURACY**

Risks in reliance on models are manageable and model analysis methods provide a level of assurance of model correctness and accuracy. In the transformation vision, models become the “language” for system development: requirements are expressed as functional models in SysML, UML or similar conventions, specifications are expressed as state-transition models (or other method of computation, as appropriate to the system), the specification models automatically produce code and component performance specifications, automatic methods generate FMEA, and high-fidelity system operation models are used for virtual testing and design of experiments for physical testing.

Using incomplete or incorrect models at any stage exposes the development program to the risk of erroneous design decisions. In the “As Is” document-centric development process, the program has greater exposure to this risk because it lacks the ability to test-and-verify the implications of requirements, specifications, and design, and because the opportunity for human error is introduced at each state. Model-centric development reduces this risk as a metamodel, as shown in Figure 33 can formalize the required information and relationships need to ensure a complete analysis. As a cadre of engineers emerges with expertise in MCE with experience using improved modeling tools and

frameworks, and model checking methods and tools, the risks from incomplete or incorrect models will decline.

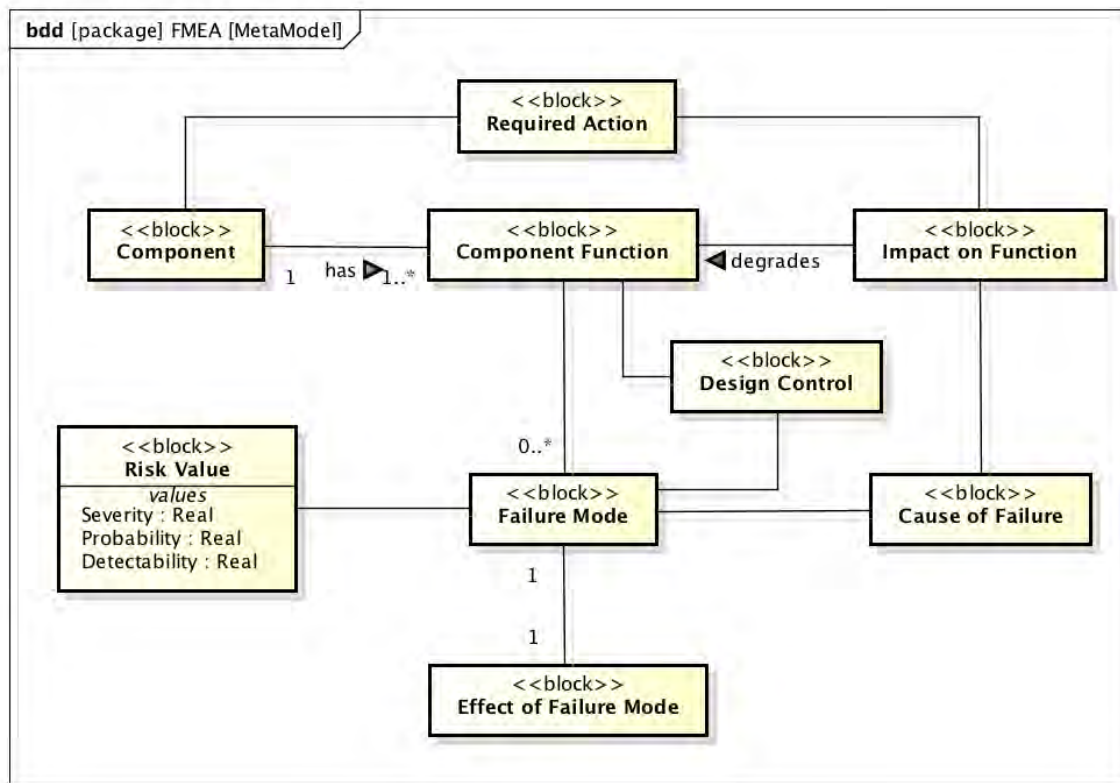


Figure 33. Example Failure Modes and Effects Analysis Metamodel

Using formal modeling methods based on a meta-model will help ensure that the system models can be checked for completeness and correctness as they are being developed. Automated model checking methods procedures and tools is a growing field with commercial products and expanding scope. These tools automatically check model completeness, and, to some extent, correctness.

Model-centric development introduces a different risk – the risk of uncritical review of the modeling and analysis methods and results. In document-centric development there is usually health skepticism, and reliance on experienced subject matter experts to review the development documents. In model-centric development, DoD will need to develop a cadre of experts with expertise both in the domain and in modeling and analysis methods.

## ERROR AND UNCERTAINTY SOURCE AND PROPAGATION ANALYSIS

Error and uncertainty propagation analysis is a developing area of research and development. These methods, procedures and tools assess the accuracy of models, and put “error bars” on the outputs. This information will inform decision makers with a realistic confidence in model-based decisions, and will inform the modelers of the need for refined modeling and calibration.

The goals of uncertainty propagation are to quantify the accuracy of model outputs, and to identify the contributing model elements. Many methods are currently in use or being researched that can propagate error through a system, including (1) nondeterministic analysis via brute force (such as Monte Carlo (MC)), (2) univariate dimension reduction, (3) deterministic model composition, (4) error budgets, (5) interval

analysis, (6) Bayesian inference, (7) anti-optimizations, and (8) error propagation via Taylor-series expansion. This is an ongoing area of research.

NASA Ames identified discretization of signals and discretization of time as a source of error in cyber-physical systems, that was particularly important in flight-critical systems since the effects can have non-deterministic effects on control system behavior. Bhatt et al. developed a model-based analysis method to propagate the errors associated with signal type and range bounds through the model to analyze the possible effects of the errors on the cyber-physical system's behavior [12]. They demonstrate the run time and scalability of the proposed approach on a set of avionics models developed for a commercial aircraft.

For certification requirements, structural loads of aircraft have to be demonstrated with required margins. These structural loads can be predicted by calculations. Estimations of these loads can only be checked and confirmed with real flight measurements. But these estimations are also subjected to uncertainties. The most common method is to measure the effects of flight loads on the structure during flight, which means measuring local strains. Another method is by measuring pressure distribution during flight. There are uncertainties in both the strain (or pressure) measurements, the location of the measurement points, and the loads. Gonzalez et al. present an uncertainty quantification method for flight load estimation that accounts for the multiple error sources [56].

---

## RISK IN A COLLABORATIVE ENVIRONMENT

Risk is not limited just to NAVAIR, it must be considered during the interactions with contractors in a continuous way rather than the monolithic reviews, especially in the context of a “radical transformation.” For example, can we create a means to enable NAVAIR to continuously use model measures as an assessment of the design and risk of a continuously evolving contractor's design/system rather than having document-based reviews? If so, then:

- There might be a need for new types of policies or governance
- It has been suggested that there may need to be some type of a policy reference model that:
  - Provides a common way to guide the use of artifacts to make decisions
  - Identify evidence (derived from models)
  - Access information
  - Analyze information leading to a Decision
    - With quantification of uncertainties
- Some “radical” transformation thoughts
  - A continual assessment of the model (all of the models) maturity
    - Possibly with different Model Maturity Levels (MML)
  - If the models cover all aspects of the aircraft
    - Related to the reference architecture/model of an aircraft
  - We can have a cumulative quantity that represents the state of the design and a measure of risk (uncertainty relative to our understanding of the margins)

It must be stated that the above scenarios about “eliminating reviews” through the use of model measures does not eliminate the interaction between NAVAIR and contractors, rather we suggest that there is a need for continuous collaboration among all stakeholders and those interactions can be done on a weekly basis or in a more workshop-based approach in the context of models. In this approach, could a “radical transformation” in the way that government and contractors interact reduce risk? One of the organizational discussions reflected on this concept of continuous collaboration using model; the

following is a true and positive story related to a Navy customer in the use of Simulink modeling to overcome numerous issues:

- The contractor started this interaction, because they were several years behind, and had not made any “real” progress
- They started modeling, which uncovered many requirement errors/issues
- This helped them understand the complexity and realized the effort was much more extensive than they had originally estimated
- Started open discusses with their customer (Navy)
  - Models provided tangible technical information about the problem
  - After a little explanation about the modeling approach, the customer was able to understand the models
  - They both realized that requirement shall statements cannot provide the needed information, and many were just wrong (incorrect, contradictory, or not what the Navy wanted)
  - Documented issues directly in the models
  - Realized that the models were in a constantly changing state, but the contractor built a trust relationship with the Navy understanding that the models were in a continuously evolving state
  - Each passing week they would review the models and could reflect on the issues that were recorded in the models

There are other variants of the operational model that were recently discussed by NASA/JPL in the way that they use models and reviews in a different way than the traditional “gate” reviews (e.g., SRR, SFR) in a model-centric way [35].

---

## RISK RELATED RESEARCH

SERC research teams are involved in several related research efforts that will be leveraged in the risk framework. We need to explore how the following can be leveraged:

- Trust under Uncertainty - Quantitative Risk; SERC RT-107 [123].
- The High Performance Computing Modernization (HPCM) CREATE [98] program to use high-fidelity models in systems design is establishing a working group on Uncertainty Quantification. SERC partners are collaborating with NAVSEA and the HPCM program.
- The DARPA internet-fabrication (iFab) project sponsored research by a SERC collaborator to develop software to automatically detect and complete gaps in specifications for a “build to” design.
- The US Army TARDEC is developing knowledge models to capture design factors and relationships in system design and development. The resulting decision breakdown structure and process should help distinguish substantive design and engineering decisions versus artifacts of the “As Is” process. SERC partners are coordinating with this effort [52] [117].
- OSD is sponsoring a SERC RT-142 Quantitative Technical Risk project to identify “risk leading indicators” and “risk estimating relationships,” analyzing the consistency, completeness, and complexity of the system architecture, requirements, task structure, and team organization, and combining those with TRL/IRL levels and Advancement Degree of Difficulty indicators (this project is being conducted in collaboration with TARDEC and an acquisition program).
- The Engineered Resilient Systems (ERS) [63] effort is addressing lost opportunity by making early concept & design decisions, the time and cost to reverse decisions, and tradeoffs between

timely but bad decisions versus deferred decisions. SERC partners are collaborating with the NAVSEA ERS and set-based design projects.

## PART II SUMMARY

---

Our research suggests that model-centric engineering is in use and adoption seems to be accelerating. Model-centric engineering can be characterized as an overarching digital approach for integrating different model types with simulations, surrogates, systems and components at different levels of abstraction and fidelity across disciplines throughout the lifecycle. Organizations are progressing beyond model-based to model-centric where integration of computational capabilities, models, software, hardware, platforms, and humans-in-the-loop allows us to assess system designs using dynamic models and surrogates to support continuous and often virtual verification and validation in the face of changing mission needs.

Enabling digital technologies are changing how organizations are conceptualizing, architecting, designing, developing, producing, and sustaining systems and systems of systems (SoS). There are many enablers that relate to characteristics of a holistic approach (this list is not exhaustive):

- Mission-level simulations that are being integrated with system simulation, digital assets & products providing a new world of services
- Leaders are embracing change and adapting to use digital strategies faster than others
- Modeling environments to create dynamic operational views (e.g., DoDAF OV-1) are increasingly used, which used to be static pictures
- 1D, 2D & 3D models have simulation and analysis capabilities (mostly physics-based) are common in practice
- Platform-based approaches with virtual integration help automakers deliver vehicle faster
- Modeling and simulation in the automotive domain is reducing the physical crash testing (e.g., from 400 to 40); this could imply that modeling and simulation can reduce test flights, which are very costly as it is difficult to get flight clearances on air craft that have advanced new capabilities
- Design optimization and trade study analysis allows for more systematic design of experiments and allows engineering to make many more excursions through the design space
- Engineering affordability analysis is a risk-based approach that could be used to significantly reduce flight tests by focusing on those flights that have the most uncertainty about margins of performance
- Risk modeling and analysis
- Pattern-based modeling based on ontologies with model transformation and analysis
- Domain-specific modeling languages
- Set-based design
- Modeling and simulation of manufacturing

This list is not exhaustive. This report provides scenarios and examples. There are emerging ideas that are envisioned to come into play within the 10-year timeframe of NAVAIR's future state, such as:

- Computer augmentation, where digital assistance will begin to understand what we are trying to model and through advances such as machine learning and integrated visualization can act as a knowledge librarian helping us to model some aspects of the problem or solution at an accelerating pace
- Explosion of interactive visualization, which we will need as we have a "sea" of data and information derived from a "sea" of models with HPC computing capabilities

## APPENDIX A: FACTOR DEFINITIONS

The following is the current set (fifth version) of the set of factors associated with the discussion measurement instrument. As the discussions with organizations are held, these factors and the associated categories will be refined.

**Table 2. Discussion Instrument Factor Definition**

Factor Category	Factors	<u>General</u> These factors relate to the degree to which advance MBSE provides a holistic approach to SE	Commentary
Magnitude of applicability over the lifecycle	Organizational Scope	What is the scope of the MBSE usage? Normally, when thinking about NAVAIR systems the scope is quite large and involves large programs. Therefore, what organizational scope does the MBSE usages apply: Program, Project, an entire Business Unit, Platform (e.g., a type of a specific aircraft, tank, automobile), Department, or Site.	Key to all of these questions is that we are looking for "Technical Feasibility of "doing everything with model." We recognize that actual adoption can be difficult, and might not make sense on older systems. Therefore related to this, it is probably best to ensure that the question perspectives come from a Chief Engineer, Chief Technical Offer, MBSE Organizational Specialist and possibly the Program Manager. To carry this a step further, it might also be important to keep the "systems" perspective in mind, because some of the concepts discussed may have been applied in Hardware and possibly in Software (e.g., the control laws for the F35 are built in Simulink, with auto code generation, and a significant portion of auto test generation), but not completely at the Systems level.
	Scope Impact	How broadly does the answers cover the entire lifecycle (for example, a university research project might be very advanced in terms of analysis or simulation, but it does not cover the entire DoD 5000 lifecycle).	The answer to this question has a lot of weight, because we need to consider answer in context of lifecycle applicable to NAVAIR (and in general DoD Acq. Programs).
Proven beyond a research concept	Demonstrations	Are the capabilities discussed actually in operations - have they been demonstrated?	We want to understand that things discussed are more than just research concepts.
Crossing the Virtual V	Integrated Simulation	In order to Cross the Virtual V, there will be many types of modeling and simulation required to support various type of domains within the system. To what degree are the simulations integrated, and better yet do different simulations work off of shared models?	In order to "cross the virtual V" during the early stages of development, it is important to understand if the inputs/outputs from one set of simulations can feed another (e.g., to be able to understand the capability in the mission context)
	Formal Analysis	Are the analyses (e.g., property analysis) formal, meaning that they are performed on models automatically?	Is the analysis fully automated from the models (H) or is there human interpretation required (M or L) or none (L)?



	Domain Specific	Are the different types of models related to the domain? For example, control system engineers often use Simulink/Matlab. Also, most modeling and simulation environments are domain-specific.	Domain-specific modeling languages are an emerging trend; these types of approaches provide intuitive abstractions (often graphical) that are familiar to engineers within the domain. Rather, SysML, while good for systems engineers, it might not be applicable to flight controls, networks, fluid dynamics, etc. In addition, there is not significant support for automated V&V from SysML as the semantics are not very rich.
Cross Domain Coverage	Domain Interoperability	Are the models that are in different, but related domains integrated? Are the models consistent across the domains?	For example, are the models that are used for performance the same models used for integrity/dependability analysis?
	Synthesis/Generation	Can the models be used for synthesis/generation of other related artifacts such as code, simulation, analysis, tests and documentation	
	Meta-Model/Model Transformations	Are the models used in one domain, or for one purpose, transformable into another domain where the well-defined semantics in one domain is carried through the transformation into the other domain; if so are they known to be consistent?	We know that one type of modeling is not always appropriate for everything, and that is why there is emergence of Domain-Specific Modeling languages; the key question is: are the models for one use consistent for other users (e.g., performance, integrity).
Virtual System Representation	Surrogate Integration	Are surrogates used to support analysis, and are the results of the surrogates captured so that they can be factored into modeling and simulation in the future?	Example, Formula 1 racing, uses data logging during physical experimentation and then factors results (and logs) back into simulation environment; can we fly some new capability on an existing aircraft and then factor the results from the test flights back into the modeling and simulation environments? This in the future should allow more virtual flight testing (once the approach becomes trusted).
	Formal Capability Assessment	How well do the models, simulations and analyses capabilities support the ability to understand the capabilities being developed?	Are the abstractions from the models still "rich enough" to be representative of the actual mission environment when used in a virtual environment? For example, if we use a 3D immersive environment, can we understand the physical characteristic of the operational system?
	Virtual Accuracy/Margin Analysis	Are the modeling, simulation and analysis accurate? How well do they allow the designers to understand the margins?	As an example, margin tolerances at the component level can propagate as the system is composed (or assembled). Are these factors understood and controlled?
	3D Immersive Environments	What is the degree to which 3D Immersive Environments are used to improve the understanding (and possibly training) of the virtual systems.	

Management Criticality Risks (Relates to Task 4)	Risk Management	Is there proper risk management identification, analysis and mitigations applied based on the use of models?	This should also consider: - Adequately deal with critical timelines - Integrated operational risk - Change management (model-based change management is different than document-based)
	Predictive Analytics	Are there models used to support a quantitative approach to risk management?	
Attributes of Modeling Maturity	Model-based metrics	Are there model-based metrics (or a comprehensive set of model measurements) and are they used to support the management of programs/projects?	The use of model-based metrics reflects on the modeling maturing of the organization.
	Multi-model interdependencies / consistency and semantic precision	If the organization is dealing with many different types of models, are the interdependencies managed and are the models semantically precise enough to manage model consistency?	Dealing with interdependencies and modeling consistency often deals with having a detailed understanding of the semantics across models. Positive results for this answer suggest a very advanced use of models.
	High Performance Computing (HPC)	Is HPC applied to the modeling, simulation and analysis efforts?	Use of HPC is an indicator of high modeling maturity.
Operational Risks (Relates to Task 4)	Procedures	Are the procedures for using the models understood, so that we can trust the model outputs to support other related types of analysis, both in terms of technical as well as risk?	This applies heavily in airworthiness (e.g., Mil Std. 516)
	Staff and Training	With the advances in the technologies associated with models, are the staff and training in place to support the use of models?	This is another indicator of a more advanced organization. As a side effect the use of 3D Immersive systems can be valuable in both collaboration and early training.
	Human Factors	How well are human factors supported by the modeling, simulation and analysis capabilities? This should consider Usability.	
Indirect support from Models	Certification	How well do the models-based automation and practices support certifications (if required)?	If not applicable use M - for Medium
	Regulation	How well do the models-based automation and practices support regulations (if required)?	If not applicable use M - for Medium
	Modeling and Simulation Qualification	How much do we trust our models?	

## ACRONYMS AND ABBREVIATION

---

This section provides a list of some of the terms used throughout the paper. The model lexicon should have all of these terms and many others.

AADL	Architecture Analysis & Design Language
ACAT	Acquisition Category
AFT	Architecture Framework Tool of NASA/JPL
AGI	Analytical Graphics, Inc.
AGM	Acquisition Guidance Model
ANSI	American National Standards Institute
AP233	Application Protocol 233
ATL	ATLAS Transformation Language
ASR	Alternative System Review
AVSI	Aerospace Vehicle Systems Institute
BDD	SysML Block Definition Diagram
BN	Bayesian Network
BNF	Backus Naur Form
BOM	Bill of Material
BPML	Business Process Modeling Language
CAD	Computer-Aided Design
CASE	Computer-Aided Software Engineering
CDR	Critical Design Review
CEO	Chief Executive Officer
CESUN	International Engineering Systems Symposium
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CORBA	Common Object Requesting Broker Architecture
CREATE	Computational Research and Engineering for Acquisition Tools and Environments
CWM	Common Warehouse Metamodel
dB	Decibel
DBMS	Database Management System
DAG	Defense Acquisition Guidebook
DARPA	Defense Advanced Research Project Agency
DAU	Defense Acquisition University
DCDR	Digital design from Critical Design Review (CDR)
DL	Descriptive Logic
DoD	Department of Defense
DoDAF	Department of Defense Architectural Framework
DoE	Design of Experiments
DSL	Domain Specific Languages
DSM	Domain Specific Modeling
DSML	Domain Specific Modeling Language
E/DRAP	Engineering Data Requirements Agreement Plan
ERS	Engineered Resilient Systems
FAA	Federal Aviation Administration
FMEA	Failure Modes and Effects Analysis
FMI	Functional Mockup Interface
FMU	Functional Mockup Unit

GAO	Government Accounting Office
HPC	High Performance Computing
HPCM	High Performance Computing Modernization
HW	Hardware
I&I	Integration and Interoperability
IBM	International Business Machines
IBD	SysML Internal Block Diagram
ICD	Interface Control Document
ICTB	Integrated Capability Technical Baseline
IDEFO	Icam DEFinition for Function Modeling
IEEE	Institute of Electrical and Electronics Engineers
INCOSE	International Council on Systems Engineering
IPR	Integration Problem Report
IRL	Integration Readiness Level
ISEF	Integrated System Engineering Framework developed by Army's TARDEC
ISO	International Organization for Standardization
IT	Information Technology
IWC	Integrated Warfighter Capability
JEO	Jupiter Europa Orbiter project at NASA/JPL
JSF	Joint Strike Fighter
JPL	Jet Propulsion Laboratory of NASA
Linux	An operating system created by Linus Torvalds
LOC	Lines of Code
M&S	Modeling and Simulation
MARTE	Modeling and Analysis of Real Time Embedded systems
MATRIXx	Product family for model-based control system design produced by National Instruments; Similar to Simulink
MBEE	Model-based Engineering Environment
MBSE	Model-based System Engineering
MBT	Model Based Testing
MC/DC	Modified Condition/Decision
MCE	Model-centric engineering
MDA®	Model Driven Architecture®
MDD™	Model Driven Development
MDE	Model Driven Engineering
MDSD	Model Driven Software Development
MDSE	Model Driven Software Engineering
MIC	Model Integrated Computing
MMM	Modeling Maturity Model
MoDAF	United Kingdom Ministry of Defence Architectural Framework
MOE	Measure of Effectiveness
MOF	Meta Object Facility
MOP	Measure of Performance
MVS	Multiple Virtual Storage
NASA	National Aeronautics and Space Administration
NAVAIR	U.S. Navy Naval Air Systems Command
NAVSEA	U.S. Naval Sea Systems Command
NDA	Non-disclosure Agreement
NDIA	National Defense Industrial Association

NEAR	Naval Enterprise Architecture Repository
NPS	Naval Postgraduate School
OCL	Object Constraint Language
OMG	Object Management Group
OO	Object oriented
OSD	Office of the Secretary of Defense
OSLC	Open Services for Lifecycle Collaboration
OV1	Operational View 1 – type of DoDAF diagram
OWL	Web Ontology Language
PDM	Product Data Management
PDR	Preliminary Design Review
PES	Physical Exchange Specification
PIA	Proprietary Information Agreement
PIM	Platform Independent Model
PLM	Product Lifecycle Management
POR	Program of Record
PRR	Production Readiness Review
PSM	Platform Specific Model
QMU	Quantification of Margins and Uncertainty
RT	Research Task
RFP	Request for Proposal
ROI	Return On Investment
SAVI	System Architecture Virtual Integration
SE	System Engineering
SERC	Systems Engineering Research Center
SETR	System Engineering Technical Review
Simulink/Stateflow	Product family for model-based control system produced by The Mathworks
SCR	Software Cost Reduction
SDD	Software Design Document
SE	System Engineering
SFR	System Functional Review
SLOC	Software Lines of Code
SME	Subject Matter Expert
SOAP	A protocol for exchanging XML-based messages – originally stood for Simple Object Access Protocol
SoS	System of System
Software Factory	Term used by Microsoft
SRR	System Requirements Review
SRS	Software Requirement Specification
STOVL	Short takeoff and vertical landing
SVR	System Verification Review
SW	Software
SysML	System Modeling Language
TARDEC	US Army Tank Automotive Research
TBD	To Be Determined
TRL	Technology Readiness Level
TRR	Test Readiness Review
UML	Unified Modeling Language
XMI	XML Metadata Interchange

XML	eXtensible Markup Language
US	United States
XSLT	eXtensible Stylesheet Language family (XSL) Transformation
xUML	Executable UML
Unix	An operating system with trademark held by the Open Group
UQ	Uncertainty Quantification
VHDL	Verilog Hardware Description Language
V&V	Verification and Validation
VxWorks	Operating system designed for embedded systems and owned by WindRiver

## TRADEMARKS

---

Analysis Server is a registered trademark of Phoenix Integration, Inc.

Astah SysML is Copyright of Change Vision, Inc.

BridgePoint is a registered trademark of Mentor Graphics.

Cameo Simulation Toolkit is a registered trademark of No Magic, Inc.

CORE is a registered trademark of Vitech Corporation.

IBM™ is a trademark of the IBM Corporation

iGrafx is a registered trademark of iGrafx, LCC.

Java™ and J2EE™ are trademark of SUN Microsystems

Java is trademarked by Sun Microsystems, Inc.

LDRA is a registered trademark of Trademark of LDRA Ltd. and Subsidiaries.

Linux is a registered trademark of Linux Mark Institute.

Mathworks, Simulink, and Stateflow are registered trademarks of The Mathworks, Inc.

MagicDraw is a trademark of No Magic, Inc.

MATRIXx is a registered trademark of National Instruments.

Microsoft®, Windows®, Windows NT®, Windows Server® and Windows Vista™ are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

ModelCenter, is a registered trademark of Phoenix Integration, Inc.

Modelica® is a registered trademark of the Modelica Association.

Object Management Group (OMG): OMG's Registered Trademarks include: MDA®, Model Driven Architecture®, UML®, CORBA®, CORBA Academy®, XMI®

OMG's Trademarks include, CWM™, Model Based Application Development™, MDD™, Model Based Development™, Model Based Management™, Model Based Programming™, Model Driven Application Development™, Model Driven Development™

Model Driven Programming™, Model Driven Systems™, OMG Interface Definition Language (IDL)™, Unified Modeling Language™, <<UML>>™

OMG®, MDA®, UML®, MOF®, XMI®, SysML™, BPML™ are registered trademarks or trademarks of the Object Management Group.

Oracle and Java are registered trademarks of Oracle, Inc. and/or its affiliates.

ParaMagic is a registered trademark of InterCAX, Inc.

PHX ModelCenter is a registered trademark of Phoenix Integration, Inc.

PowerPoint is a registered trademark of Microsoft, Inc.

Real-time Studio Professional is a registered trademark of ARTiSAN Software Tools, Inc.

Rhapsody is a registered trademark of Telelogic/IBM.

Rose XDE is a registered trademark of IBM.

SCADE is copyrighted to Esterel Technologies.



Simulink is a registered trademark of The MathWorks.  
Stateflow is a registered trademark of The MathWorks.  
Statemate is a registered trademark of Telelogic/IBM.  
STK is a registered trademark of Analytical Graphics, Incorporated (AGI), Inc.  
UNIX is a registered trademark of The Open Group.  
VAPS is registered at eGENUITY Technologies.  
VectorCAST is a registered trademark of Vector Software, Inc.  
Visio is a registered trademark of Microsoft, Inc.  
VxWorks is a registered trademark of Wind River Systems, Inc.  
Windows is a registered trademark of Microsoft Corporation in the United States and other countries.  
XML™ is a trademark of W3C  
All other trademarks belong to their respective organizations.

## REFERENCES

---

- [1] Ackoff, R., L. and Sheldon Rodin. *Redesigning Society*. Stanford: Stanford University Press, 2003.
- [2] Adams, B., Adam Stephens, *Dakota Sensitivity Analysis and Uncertainty Quantification, with Examples*, SNL 6230 Course on UQ/SA, April 23, 2014.
- [3] Allen, G., F. Hartman, F. Mullen, *Dynamic Multi-level Modeling Framework, Results of the Feasibility Study*, NDIA, October 2013.
- [4] ARTEMIS-GB-2012-D.46 – Annex 2, 2013. <https://ec.europa.eu/research/participants/portal/desktop/en/opportunities/fp7/calls/artemis-2013-1.html>
- [5] Baitech, L., Randall C. Smith, *Physiological Correlates of Spatial Perceptual Discordance in a Virtual Environment*, General Motors Research & Development Center Virtual Environments Laboratory.
- [6] Bankes, S., D. Challou, D. Cooper, T. Haynes, H. Holloway, P. Pukite, J. Tierno, C. Wentland, *META Adaptive, Reflective, Robust Workflow (ARROW), Phase 1b Final Report*, TR-2742, October, 2011.
- [7] Bapty, T., S. Neema, J. Scott, *Overview of the META Toolchain in the Adaptive Vehicle Make Program*, Vanderbilt, ISIS-15-103, 2015.
- [8] Bayer, Todd J., Matthew Bennett, Christopher L. Delp, Daniel Dvorak, J. Steven Jenkins, and Sanda Mandutianu. "Update - Concept of Operations for Integrated Model-Centric Engineering at JPL," 1–15. IEEE, 2011. doi:10.1109/AERO.2011.5747538.
- [9] Bayer, Todd, Seung Chung, Bjorn Cole, Brian Cooke, Frank Dekens, Chris Delp, I. Gontijo, et al. "11.5.1 Early Formulation Model-Centric Engineering on NASA's Europa Mission Concept Study." *INCOSE International Symposium* 22, no. 1 (July 2012): 1695–1710. doi:10.1002/j.2334-5837.2012.tb01431.x.
- [10] Bergenthal, J., *Final Report on the Identification of Modeling and Simulation Capabilities by Acquisition Life Cycle Phases*, Johns Hopkins University/Applied Physics Laboratory, 16th Annual Systems Engineering Conference, October, 2013.
- [11] Bergenthal, J., J. Coolahan, *Final Report on the Identification of Modeling and Simulation Capabilities by Acquisition Life Cycle Phases*, NDIA Systems Engineering Division Meeting, February 2014.
- [12] Bhatt, D., K. Schloegel, G. Madl, D. Oglesby. *Quantifying Error Propagation in Data Flow Models*. 20th Annual IEEE International Conference and Workshops on the Engineering of Computer Based Systems. 2013.
- [13] Blackburn, M.R., *What's Model Driven Engineering (MDE) and How Can it Impact Process, People, Tools and Productivity*, Systems and Software Consortium, Technical Report SSCI-2008002-MC, September, 2008 [http://www.knowledgebytes.net/downloads/Whats\\_MDE\\_and\\_How\\_Can\\_it\\_Impact\\_me.pdf](http://www.knowledgebytes.net/downloads/Whats_MDE_and_How_Can_it_Impact_me.pdf).
- [14] Blackburn, M.R., *Model-Driven Verification and Validation*, Safe & Secure Systems & Software Symposium, June, 15-17 2010. Modified from Paul Eremenko, *META Novel Methods for Design & Verification of Complex Systems*, December 22, 2009.
- [15] Blackburn, M., A. Pyster, R. Dillon-Merrill, T. Zigh, R. Turner, *Results from Applying a Modeling and Analysis Framework to an FAA NextGen System of Systems Program*, NDIA, October, 2013.
- [16] Blackburn, M., A. Pyster, R. Dillon-Merrill, T. Zigh, R. Turner, *Modeling and Analysis Framework for Risk-Informed Decision Making for FAA NextGen*, INCOSE, June 2013.
- [17] Blackburn, M., A. Pyster, R. Dillon-Merrill, T. Zigh, R. Turner, *Using Bayesian Networks for Modeling an Acquisition Decision-Making Process for the FAA NextGen Systems of Systems*, NDIA, October, 2012.
- [18] Blackburn, M., R. Busser, A. Nauman, and T. Morgan. "Life Cycle Integration Use of Model-Based Testing Tools," 2:10.D.4–1 – 10.D.4–13. IEEE, 2005. doi:10.1109/DASC.2005.1563402.
- [19] Blackburn, M., R. Busser, H. Graves, *Guidelines for Automated Analysis of System Models*, Software Productivity Consortium Technical Report, December, 2000.
- [20] Blackburn, M., Cloutier, R., Hole, E., Witus, G., 2014. *Introducing Model-Based Systems Engineering Transforming System Engineering through Model-Based Systems Engineering* (Technical Report No. TR-044). Systems Engineering Research Center.
- [21] Blackburn, Mark, Robert Cloutier, Eirik Hole, and Gary Witus. *Introducing Model-Based Systems Engineering Transforming System Engineering through Model-Based Systems Engineering*. Technical Report. Systems Engineering Research Center, March 31, 2014. <http://www.sercuarc.org/library/view/58>.

- [22] Blackburn, M., R. Cloutier, G. Witus, E. Hole, M. Bone, Transforming System Engineering through Model-Centric Engineering, SERC-2014-TR-044-2, January, 2015.
- [23] Blackburn, M., P. Denno, Virtual Design and Verification of Cyber-physical Systems: Industrial Process Plant Design, Conference on Systems Engineering Research, March, 2014; <http://dx.doi.org/10.1016/j.procs.2014.03.006>.
- [24] Blackburn, M., P. Denno, Using Semantic Web Technologies for Integrating Domain Specific Modeling and Analytical Tools, Complex Adaptive Systems Conference, Nov. 2015.
- [25] Blackburn, M., S. Kumar, Evolving Systems Engineering through Model Driven Functional Analysis, NDIA System Engineering Conference, October 2009.
- [26] Bleakley, G., A. Lapping, A. Whitfield, Determining the Right Solution Using SysML and Model Based Systems Engineering, (MBSE) for Trade Studies, INCOSE International Symposium, June, 2011.
- [27] Boehm, B., Software Cost Estimation with Cocomo II, Prentice Hall, 2000.
- [28] Box, George E. P. Empirical Model-Building and Response Surfaces. Wiley Series in Probability and Mathematical Statistics. New York: Wiley, 1987.
- [29] Brat, Guillaume, V & V of Flight-Critical Systems, NASA ARCS5 - Safe & Secure Systems & Software Symposium, June 2010.
- [30] Broy, M., M. Feilkas, M. Herrmannsdoerfer, S. Merenda, and D. Ratiu. "Seamless Model-Based Development: From Isolated Tools to Integrated Model Engineering Environments." Proceedings of the IEEE 98, no. 4 (April 2010): 526–45. doi:10.1109/JPROC.2009.2037771.
- [31] Business Process Modeling Notation. Retrieved March 2010, from Wikipedia, The Free Encyclopedia: [http://en.wikipedia.org/wiki/Business\\_Process\\_Modeling\\_Notation](http://en.wikipedia.org/wiki/Business_Process_Modeling_Notation).
- [32] Browne, D., R. Kempf, A. Hansena, M. O'Neal, W. Yates, Enabling Systems Modeling Language Authoring in a Collaborative Web-based Decision Support Tool, Conference on System Engineering Research (CSER), March, 2013.
- [33] Castet, Jean-Francois, Matthew L. Rozek, Michel D. Ingham, Nicolas F. Rouquette, Seung H. Chung, J. Steven Jenkins, David A. Wagner, and Daniel L. Dvorak. "Ontology and Modeling Patterns for State-Based Behavior Representation." American Institute of Aeronautics and Astronautics, 2015. doi:10.2514/6.2015-1115.
- [34] Chilenski, J., SAVI Principal Investigator, Don Ward, TEES SAVI Program Manager, NDIA M&S Subcommittee – Arlington, Virginia 8 April 2014.
- [35] Cooke, B., MBSE on Europa Clipper, NASA/JPL Symposium and Workshop on Model-Based Systems Engineering, January 2015.
- [36] Coolahan, J. A Vision for modeling and simulation at APL, Johns hopkins APL Technical Digest, Volume 26, number 4 (2005).
- [37] Cloutier, Robert & Mary Bone. 2015. MBSE Survey. INCOSE IW 2015. Los Angeles, CA.
- [38] Dahmann, J., BA. Aumber, M, Kelley, Importance of Systems Engineering in Early Acquisition, MITRE Corporation. Approved for Public Release; Distribution Unlimited Case # 09-0345.
- [39] DARPA, Producible Adaptive Model-based Software (PAMS) technology to the development of safety critical flight control software. PAMS has been developed under the Defense Advanced Research Projects Agency (DARPA) Disruptive Manufacturing Technologies program. Contract # N00178-07-C-2011, <http://www.isis.vanderbilt.edu/projects/PAMS>.
- [40] Darwiche, A., Modeling and Reasoning with Bayesian Networks, Cambridge University Press, 2009.
- [41] Davidoff, S., Visualization of Model Content and Engineering Process, NASA/JPL Symposium and Workshop on Model-Based Systems Engineering, January 2015.
- [42] Defense Acquisition University, Defense Acquisition Guidebook Chapter 4 – Systems Engineering, May 2013; <https://acc.dau.mil/dag4>.
- [43] Delp, C., D. Lam, E. Fosse, and Cin-Young Lee. "Model Based Document and Report Generation for Systems Engineering," 1–11. IEEE, 2013. doi:10.1109/AERO.2013.6496926.
- [44] Department of Defense, INSTRUCTION – INTERIM, NUMBER 5000.02 November 26, 2013.
- [45] Department of Defense, MIL-HDBK-516B, Department Of Defense Handbook: Airworthiness Certification Criteria, Feb, 2008; [http://www.everyspec.com/MIL-HDBK/MIL-HDBK-0500-0599/MIL-HDBK-516B\\_CHANGE-1\\_10217](http://www.everyspec.com/MIL-HDBK/MIL-HDBK-0500-0599/MIL-HDBK-516B_CHANGE-1_10217).
- [46] Department of Defense, Risk Management Guide For Dod Acquisition, Sixth Edition, August, 2006.
- [47] Elele, J.N., Assessing Risk Levels of Verification, Validation, and Accreditation of Models and Simulations,

- International Test and Evaluation Association (ITEA) Journal 2008.
- [48] DO-178B/ED-12B - Software Considerations in Airborne Systems and Equipment Certification, Radio Technical Corporation for Aeronautics Special Committee 167 (RTCA) December, 1992.
  - [49] Evans, B., Modeling and Simulation Applied in the F-35 Program, Barry Evans Lockheed Martin Aeronautics, 2011.
  - [50] Firesmith, D., Are Your Requirements Complete?, Journal of Object Technology, Volume 4, no. 1 (January 2005), pp. 27-43, doi:10.5381/jot.2005.4.1.c3.
  - [51] Flager, F., John Haymaker, A Comparison of Multidisciplinary Design, Analysis and Optimization Processes in the Building Construction and Aerospace, Stanford, December 2009.
  - [52] Graf, L., Transitioning Systems Engineering Research into Programs and Practice, NDIA 17th SE Annual Conference, October 2014.
  - [53] GAO, Problems Completing Software Testing May Hinder Delivery of Expected Warfighting Capabilities, GAO-14-322: Published: Mar 24, 2014. Publicly Released: Mar 24, 2014.
  - [54] Gaignic, Pascal, Thomas Vosgien, Marija Jankovic, Vincent Tuloup, Jennifer Berquet, and Nadège Troussier, Complex System Simulation: Proposition of a MBSE Framework for Design-Analysis Integration, Procedia Computer Science 16 (January 2013): 59–68. doi:10.1016/j.procs.2013.01.007.
  - [55] Gill, Helen. "From Vision to Reality: Cyber-Physical Systems", HCSS National Workshop on New Research Directions for High Confidence Transportation CPS: Automotive, Aviation, and Rail, November 18-20, 2008.
  - [56] Gonzales, M., C. Gogu, N. Binaud, C. Espinoza, J. Morlier, and S. Quoniam. Uncertainty quantification in aircraft load calibration. 10th World Congress on Structural and Multidisciplinary Optimization. 2013.
  - [57] Hammen, D., G. Turner, JSC Engineering Orbital Dynamics Integration Model, National Aeronautics and Space Administration, December 2014.
  - [58] Hannapel, Shari, Nickolas Vlahopoulos, and David Singer. "Including Principles of Set-Based Design in Multidisciplinary Design Optimization." American Institute of Aeronautics and Astronautics, 2012. doi:10.2514/6.2012-5444.
  - [59] Hayhurst, Kelly J., Dan S. Veerhusen, John J. Chilenski, and Leanna K. Rierson. A Practical Tutorial on Modified Condition/Decision Coverage, NASA/TM-2001-210876. <http://techreports.larc.nasa.gov/ltrs/PDF/2001/tm/NASA-2001-tm210876.pdf>
  - [60] Henson Graves, H., S. Guest, J. Vermette, Y. Bijan, H. Banks, G. Whitehead, B. Ison, Air Vehicle Model-Based Design and Simulation Pilot, Lockheed Martin, 2009; available <http://www.omgwiki.org/MBSE>.
  - [61] Herring, M., D. Owens, N. Leveson, M. Ingham, and K. Weiss. Safety-Driven Model-Based System Engineering Methodology. 2007.
  - [62] Herron, J. Model-Centric Design CAD Design in Aerospace. Retrieved from [http://www.findarticles.com/p/articles/mi\\_hb078/is\\_199801/aihibm1g16938479](http://www.findarticles.com/p/articles/mi_hb078/is_199801/aihibm1g16938479), 2006.
  - [63] Holland, J., Engineered Resilient Systems (ERS) Overview, December 2013.
  - [64] Hutchinson, J., J. Whittle, M. Rouncefield, S. Kristoffersen, Empirical Assessment of MDE in Industry, Proceedings of the 33rd International Conference on Software Engineering, 2011.
  - [65] IDEFØ, Computer Systems Laboratory of the National Institute of Standards and Technology (NIST), 1993.
  - [66] International Council on Systems Engineering (INCOSE), "MBSE initiative," January 2007; <https://connect.incose.org/tb/MnT/mbseworkshop/>.
  - [67] ISO/IEC 42010:2007, Systems and Software Engineering -- Architecture Description, 2007.
  - [68] Jackson, Ethan, and Janos Sztipanovits. "Formalizing the Structural Semantics of Domain-Specific Modeling Languages." Software & Systems Modeling 8, no. 4 (September 2009): 451–78. doi:10.1007/s10270-008-0105-0.
  - [69] Jenkins, J. S., N. Rouquette, Semantically-Rigorous systems engineering modeling using SysML and OWL, 5th International Workshop on Systems & Concurrent Engineering for Space Applications, Lisbon, Portugal, October 17-19, 2012.
  - [70] Joshi, A., M. P.E. Heimdahl. Model-Based Safety Analysis of Simulink Models Using SCADE Design Verifier. Proc. 24th Digital Avionics Systems Conference. 2005.
  - [71] Khan, O., G. Dubos, J. Tirona, S. Standley, Model-Based Verification and Validation of the SMAP Uplink Processes, IEEE Aerospace Conference, 2013.
  - [72] Kim, H., Fried, D., Menegay, P., Connecting SysML Models with Engineering Analyses to Support

- Multidisciplinary System Development, American Institute of Aeronautics and Astronautics, 2012.
- [73] Kim, H., Fried, D., Menegay, P., G. Soremekun, C. Oster, Application of Integrated Modeling and Analysis to Development of Complex Systems, Conference on Systems Engineering Research, 2013; <http://dx.doi.org/10.1016/j.procs.2013.01.011>.
  - [74] Kortelainen, J., Semantic Data Model for Multibody System Modelling, Dissertation, Lappeenranta University of Technology, 2011.
  - [75] Leveson, N., A New Accident Model for Engineering Safer Systems, Safety Science, Vol. 42, No. 4, April 2004.
  - [76] Matei, I., C. Bock, SysML Extension for Dynamical System Simulation Tools, National Institute of Standards and Technology, NISTIR 7888, <http://dx.doi.org/10.6028/NIST.IR.7888>, October 2012, <http://nvlpubs.nist.gov/nistpubs/ir/2012/NIST.IR.7888.pdf>.
  - [77] McFarland, J., Uncertainty Analysis For Computer Simulations Through Validation And Calibration, Dissertation, Vanderbilt University, May 2008.
  - [78] McFarland, J., Sankaran Mahadevan, Vicente Romero, Laura Swiler, Calibration and Uncertainty Analysis for Computer Simulations with Multivariate Output, AIAA, October, 2007.
  - [79] McKelvin, Jr., Mark, and Alejandro Jimenez. "Specification and Design of Electrical Flight System Architectures with SysML." American Institute of Aeronautics and Astronautics, 2012. doi:10.2514/6.2012-2534.
  - [80] MIL-HDBK-516C, Department Of Defense Handbook: Airworthiness Certification Criteria, December 12, 2014.
  - [81] Model Based Enterprise, <http://model-based-enterprise.org/>.
  - [82] Murray, Brian T., Alessandro Pinto, Randy Skelding, Olivier L. de Weck, Haifeng Zhu, Sujit Nair, Narek Shougarian, Kaushik Sinha, Shaunak Bodardikar, and Larry Zeidner. META II Complex Systems Design and Analysis (CODA), 2011.
  - [83] NAOMI Project, Lockheed Martin Advanced Technology Laboratories; <http://www.atl.external.lmco.com/programs/STI/programs/program1.php#experimentalinfrastructure>, 2013.
  - [84] National Institute of Standards and Technology, Foundations for Innovation in Cyber-Physical Systems, Workshop Report, 2013.
  - [85] NAVAIRINST 13034.1C, Navair Instruction: Flight Clearance Policy For Air Vehicles And Aircraft Systems, September, 28, 2004.
  - [86] Navy Integration and Interoperability (I&I) Integrated Capability Framework (ICF), Operational Concept Document, Version 2.0, 30 September 2013.
  - [87] Newcomer, J. T., SANDIA REPORT, SAND2012-7912 Unlimited Release Printed September 2012, A New Approach to Quantification of Margins and Uncertainties for Physical Simulation Data. (<http://prod.sandia.gov/techlib/access-control.cgi/2012/127912.pdf>).
  - [88] Nixon, D. W., Flight Control Law Development for the F-35 Joint Strike Fighter, October 5, 2004.
  - [89] Oberkamp, William Louis, Timothy Guy Trucano, and Martin M. Pilch. "Predictive Capability Maturity Model for Computational Modeling and Simulation.," October 1, 2007. <http://www.osti.gov/servlets/purl/976951-meC28s/>.
  - [90] Object Management Group, MBSE Wiki, Ontology Action Team, <http://www.omgwiki.org/MBSE/doku.php?id=mbse:ontology>, 2014.
  - [91] Object Management Group, XML Metadata Interchange (XMI), Version, 2.4.2, April 2014, <http://www.omg.org/spec/XMI/2.4.2>.
  - [92] Object Management Group. OMG Unified Modeling Language™ (OMG UML), Superstructure. 2011. Version 2.4.1. Available from: <http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF>.
  - [93] Object Management Group. OMG Systems Modeling Language (OMG SysML™). 2012. Version 1.3. Available from: <http://www.omg.org/spec/SysML/1.3/PDF>.
  - [94] Papadopoulos, Y., D. Parker, C. Grant. A Method and Tool Support for Model-based Semi-automated Failure Modes and Effects Analysis of Engineering Designs. Proc. 9th Australian Workshop on Safety Related Programmable Systems. 2004.
  - [95] Paredis, C., Y. Bernard, R. Burkhart, D. Koning, S. Friedenthal, P. Fritzson, N. Rouquette, W. Schamai, An Overview of the SysML-Modelica Transformation Specification, INCOSE International Symposium, Chicago, IL, July, 2010.

- [96] Peak, R., S. Cimentalay, A. Scott, M. Wilson, B. Aikens, D. Martin, Verification, Validation, and Accreditation Shortfalls for Modeling and Simulation, Final Technical Report SERC-2011-TR-018, Systems Engineering Research Center, 2011.
- [97] Pearl, J. (1985). "Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning" (UCLA Technical Report CSD-850017). Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine, CA. pp. 329–334. Retrieved 2009-05-01.
- [98] Post, D., Computational Research Engineering Acquisition Tools and Environments, A DoD Program to Aid Acquisition Engineering, NDIA, October 2014.
- [99] Ray, S., G. Karsai, K. McNeil, Model-Based Adaptation of Flight-Critical Systems, Digital Avionics Systems Conference, 2009.
- [100] Rasumussen, R., R. Shishko, Jupiter Europa Orbiter Architecture Definition Process, INCOSE Conference on Systems Engineering Research, Redondo Beach, California, April 14-16, 2011.
- [101] Rhodes, D. H., A. M. Ross, P. Grogan, O. de Weck, Interactive Model-Centric Systems Engineering (IMCSE), Phase One Technical Report SERC-2014-TR-048-1, Systems Engineering Research Center, September 30, 2014.
- [102] Ressler, S., What's That 3D Model Doing in my Web Browser, Model-Based Enterprise Summit 2014, <http://math.nist.gov/~SRessler/x3dom/revealjs14/mbeNISTtalk.html#/>
- [103] Rizzo, D., M. R. Blackburn, Use of Bayesian networks for qualification planning: a predictive analysis framework for a technically complex systems engineering problem, Complex Adaptive Systems Conference, November, 2015.
- [104] Rodano, M., K. Giammarco, A Formal Method for Evaluation of a Modeled System Architecture Matthew Stevens Institute of Technology, Complex Adaptive Systems Conference, 2013.
- [105] Romero, V., Elements of a Pragmatic Approach for dealing with Bias and Uncertainty in Experiments through Predictions: Experiment Design and Data Conditioning, "Real Space" Model Validation and Conditioning, Hierarchical Modeling and Extrapolative Prediction, SAND2011-7342 Unlimited Release Printed November 2011.
- [106] Romero, V., Uncertainty Quantification and Sensitivity Analysis—Some Fundamental Concepts, Terminology, Definitions, and Relationships, UQ/SA section of invited paper for AIAA SciTech2015 Non-Deterministic Approaches Conference, Jan 5-9, 2015, Orlando, FL.
- [107] Rothenberg, J. L. E. Widman, K. A. Loparo, N. R. Nielsen, The Nature of Modeling, Artificial Intelligence, Simulation and Modeling, 1989.
- [108] SAE ARP4761. Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment. SAE International, December 1996.
- [109] Sandia National Laboratory, Dakota, <https://dakota.sandia.gov/>.
- [110] Schindel, W. D., Failure Analysis: Insights from Model-Based Systems Engineering. Proc. INCOSE Int'l Symposium. 2010.
- [111] Schroeder, C. A., A Study of How Model-Centric Engineering Relates to Time-To-Market and Agility to Accommodate Customer-Required Changes, Dissertation, Indiana State University, 2011.
- [112] Simko, Gabor, Tihamer Levendovszky, Sandeep Neema, Ethan Jackson, Ted Bapty, Joseph Porter, and Janos Sztipanovits. "Foundation for Model Integration: Semantic Backplane," 2012.
- [113] Singer, David J., Norbert Doerry, and Michael E. Buckley. "What Is Set-Based Design?: What Is Set-Based Design?" Naval Engineers Journal 121, no. 4 (October 2009): 31–43. doi:10.1111/j.1559-3584.2009.00226.x.
- [114] Snooke, N., Model-Based Failure Modes and Effects Analysis of Software. Proceedings DX04. 2004.
- [115] Spangelo, S. D. Kaslow, C. Delp, L. Anderson, B. Cole, E. Foyse, L. Cheng, R. Yntema, M. Bajaj, G. Soremekum, J. Cutler, MBSE Challenge Team, Model Based Systems Engineering (MBSE) Applied to Radio Aurora Explorer (RAX) CubeSat Mission Operational Scenarios, IEEEAC Paper #2170, Version 1, Updated 29/01/2013.
- [116] System Engineering Research Center, INCOSE, Stevens, Report Of The Workshop On The Relationship Between Systems Engineering And Software Engineering, Workshop sponsored by Stevens, INCOSE, SERC, June 2014.
- [117] Umpfenbach, E., Integrated System Engineering Framework (ISEF), NDIA Systems Engineering Conference, October 2014.
- [118] <http://www.vitechcorp.com/products/core.shtml>
- [119] Wagner, D.A., M. Bennett, R. Karban, N. Rouquette, S. Jenkins, M. Ingham, An Ontology for State Analysis:



- Formalizing the Mapping to SysML, IEEE Aerospace Conference, 2012.
- [120] Wade, J., R. Cohen, M. Blackburn, E. Hole, N. Bowen, Systems Engineering of Cyber-Physical Systems Education Program, World Innovation Summit for Education, Nov. 2015.
  - [121] West, T., A. Pyster, Untangling the Digital Thread: The Challenge and Promise of Model-Based Engineering in Defense Acquisition, INCOSE INSIGHT, Volume 18, Issue 2, pages 45–55, August 2015.
  - [122] Wikipedia, Ontology, [http://en.wikipedia.org/wiki/Ontology\\_\(information\\_science\)](http://en.wikipedia.org/wiki/Ontology_(information_science)), 2014.
  - [123] Witus, G., W. Bryzik, Trust under Uncertainty - Quantitative Risk, SERC RT-107, Systems Engineering Research Review, December, 2014.
  - [124] Witherell, Paul, Boonserm Kulvatunyou, and Sudarsan Rachuri. "Towards the Synthesis of Product Knowledge Across the Lifecycle," V012T13A071. ASME, 2013. doi:10.1115/IMECE2013-65220.
  - [125] World Wide Web Consortium. OWL 2 Web Ontology Language Document Overview. 2009. Available from: <http://www.w3.org/TR/2009/REC-owl2-overview-20091027/>.
  - [126] Xie, H. Li, X., C. Liu., The Model-Based and Bidirectional Software Failure Mode and Effect Analysis Method. IEEE Intl Conf on Reliability, Maintainability and Safety (ICRMS). 2014.
  - [127] Zentner, J., Ender, T., Ballestrini-Robinso, S., On Modeling and Simulation Methods for Capturing Emergent Behaviors for Systems-of-Systems, 12th Annual Systems Engineering Conference, October, 2009.
  - [128] Zimmerman, P., Model-Based Systems Engineering (MBSE) in Government: Leveraging the 'M' for DoD Acquisition, 2014 INCOSE MBSE Workshop January 25, 2014.
  - [129] zur Muehlen, M., D. Hamilton, R. Peak, Integration of M&S (Modeling and Simulation), Software Design and DoDAF, SERC-2012-TR-024, 2012.